CONTROL DATA CORPORATION

CONTROL DATA® CYBER 70 SERIES COMPUTER SYSTEMS MODELS 72, 73, 74 6000 SERIES COMPUTER SYSTEMS

KRONOS® 2.1 WORKSHOP REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
A	Released February 1, 1974
B	Revised May 15, 1974
C	Revised June 15, 1975
D	Revised April 15, 1976

Publication No.
97404700 D

Additional copies of this manual may be obtained from the nearest Control Data Corporation sales office.

Address comments concerning this manual to:

Control Data Corporation
Publications Department
8100 34th Avenue
Minneapolis, Minnesota 55440
Attn: PSD Publications

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	SFC †	REV
Title Page		D
ii		C
iiA		D
iiB		D
iiC		D
iiD		D
iiE		D
iiF		D
iii		B
iv		C
v		C
vi		C
vii		C
viii		C
ix		C
x		C
xi		C
xii		C
xiii		C
xiv		C
xv		C
xvi		D
xvii		D
xviii		D
xix		D
xx		D
xxi		D
xxii		D
xxiii		D
xxiv		D
xxv		D
xxvi		D
1-1		B
1-2		B
1-3		C
1-4		B
1-5		B
1-6		B
1-7		B
1-8		B
1-9		B

PAGE	SFC †	REV
1-10		C
1-11		B
1-12		C
1-13		B
1-14		C
1-15		C
2-1		B
2-2		B
2-3		A
2-4		C
2-5		C
2-6		C
2-7		C
2-8		C
2-9		C
2-10		C
2-11		C
2-12		C
2-12.1		C
2-12.2		C
2-13		C
2-14		C
2-15		C
2-16		C
2-16.1		C
2-16.2		C
2-16.3		C
2-16.4		C
2-16.5		C
2-16.6		C
2-17		C
2-18		B
2-19		C
2-20		C
2-20.1		C
2-20.2		C
2-21		C
2-22		C
2-23		B
2-24		B
2-24.1		C

PAGE	SFC †	REV
2-25		B
2-25.1		C
2-26		C
2-26.1		C
2-27		C
2-27.1		C
2-28		C
2-29		A
2-30		C
2-30.1		C
2-30.2		C
2-30.3		C
2-31		B
2-32		B
3-1		B
3-2		B
3-3		A
3-4		A
3-5		A
3-6		A
3-7		B
3-8		B
3-9		A
3-10		C
3-10.1		C
3-10.2		C
3-10.3		C
3-10.4		C
3-10.5		C
3-10.6		C
3-10.7		C
3-10.8		C
3-10.9		C
3-10.10		C
3-10.11		C
3-10.12		C
3-10.13		C
3-11		B
3-12		B
3-13		B
3-14		B

PAGE	SFC †	REV
3-15		C
3-16		B
3-17		C
3-18		A
3-19		A
3-20		B
3-21		C
3-21.1		C
3-22		C
3-23		C
3-24		C
3-25		B
3-26		B
3-27		B
3-28		C
3-29		A
3-30		C
3-30.1		C
3-30.2		C
3-30.3		C
3-30.4		C
3-30.5		C
3-30.6		C
3-30.7		C
3-30.8		C
3-30.9		C
3-30.10		C
3-30.11		C
3-30.12		C
3-30.13		C
3-31		C
3-32		A
3-33		B
3-34		B
3-35		B
3-36		C
3-37		B
3-38		B
3-39		B
3-40		B
3-41		B

†SFC Software Feature Change

LIST OF EFFECTIVE PAGES (Continued)

PAGE	SFC †	REV
3-42		B
3-43		B
3-44		C
3-45		B
3-46		C
3-47		A
3-48		C
3-49		C
3-50		B
3-51		B
3-52		B
3-53		A
3-54		C
3-55		A
3-56		A
3-57		B
3-58		C
4-1		C
4-2		C
4-3		C
4-3.1		C
4-4		C
4-5		B
4-6		B
4-7		B
4-8		B
4-9		B
4-10		B
4-11		B
4-12		C
4-13		C
4-14		B
4-15		B
4-16		B
4-17		B
4-18		B
4-19		B
4-20		C
4-21		C
4-22		C
4-23		C

PAGE	SFC †	REV
4-23.1		C
4-24		C
4-25		B
4-26		B
4-27		B
4-28		B
4-29		B
4-30		B
5-1		B
5-2		A
5-3		B
5-4		A
5-5		A
5-6		C
5-7		C
5-8		C
5-9		C
5-10		B
5-11		C
5-12		C
5-13		C
5-14		A
5-15		B
5-16		B
5-17		C
5-18		B
5-18.1		C
5-18.2		C
5-18.3		C
5-18.4		C
5-18.5		C
5-18.6		C
5-19		B
5-20		B
5-21		B
5-22		B
5-23		B
5-24		A
5-25		C
5-26		C
5-27		A

PAGE	SFC †	REV
5-28		C
5-29		B
5-30		A
5-31		A
5-32		A
5-33		B
5-34		A
5-35		A
5-36		A
5-37		A
5-38		A
5-39		B
5-40		B
5-41		B
5-42		C
5-42.1		C
5-43		B
5-44		B
5-45		C
5-46		C
5-47		B
6-1		A
6-2		C
6-3		C
6-4		A
6-5		A
6-6		A
6-7		A
6-8		A
6-9		C
6-10		B
6-11		B
6-12		A
6-13		B
6-14		A
6-15		B
6-16		A
6-17		C
6-18		C
6-19		A
6-20		A

PAGE	SFC †	REV
6-21		A
6-22		A
6-23		A
6-24		A
6-25		A
6-26		A
6-27		C
6-28		B
6-29		A
6-30		B
6-31		A
6-32		A
6-33		A
6-34		B
6-35		A
6-36		A
6-37		A
6-38		A
6-39		A
6-40		C
6-41		B
6-42		B
6-43		B
6-44		C
6-45		B
6-46		B
6-47		B
6-48		B
6-49		B
6-50		B
6-51		B
6-52		B
6-53		A
7-1		A
7-2		A
7-3		C
7-4		A
7-5		B
7-6		B
7-7		A
7-8		A

†SFC Software Feature Change

LIST OF EFFECTIVE PAGES (Continued)

PAGE	SFC †	REV
7-9		B
7-10		C
7-11		A
7-12		C
7-13		B
7-13.1		C
7-14		B
7-15		B
7-16		C
7-17		C
7-18		B
7-19		C
7-20		C
7-21		C
7-22		C
7-23		C
7-24		C
7-25		C
8-1		A
8-2		A
8-3		B
8-4		A
8-5		B
8-6		A
8-7		A
8-8		A
8-9		B
8-10		C
8-11		A
8-12		C
8-13		B
8-14		A
8-15		A
8-16		B
8-17		A
8-18		B
8-19		A
8-20		A
8-21		A
8-22		C
8-23		A

PAGE	SFC †	REV
8-24		A
8-25		A
8-26		A
8-27		A
8-28		A
8-29		C
8-30		C
8-31		A
8-32		C
9-1		C
9-1.1		C
9-1.2		C
9-1.3		C
9-1.4		C
9-1.5		C
9-1.6		C
9-1.7		C
9-1.8		C
9-1.9		C
9-1.10		C
9-1.11		C
9-2		C
9-3		B
9-4		B
9-5		C
9-6		C
9-6.1		C
9-6.2		C
9-6.3		C
9-6.4		C
9-7		B
9-8		A
9-9		A
9-10		B
9-11		A
9-12		C
9-13		A
9-14		A
9-15		A
9-16		A
9-17		A

PAGE	SFC †	REV
9-18		A
9-19		A
9-20		B
9-21		A
9-22		C
9-23		C
9-24		C
9-25		C
9-26		C
9-27		C
9-28		C
9-29		C
9-30		C
9-31		C
9-32		C
9-33		C
10-1		A
10-2		C
10-2.1		C
10-2.2		C
10-2.3		C
10-2.4		C
10-2.5		C
10-2.6		C
10-3		A
10-4		B
10-5		A
10-6		B
10-7		A
10-8		B
10-9		A
10-10		A
10-11		A
10-12		B
10-13		B
10-14		A
10-15		A
10-16		A
10-17		B
10-17.1		C
10-17.2		C

PAGE	SFC †	REV
10-17.3		C
10-17.4		C
10-17.5		C
10-18		C
10-19		B
10-20		A
10-21		C
10-22		C
10-23		C
10-24		C
10-25		C
10-26		C
10-27		C
10-28		C
10-29		C
10-30		C
11-1		C
11-1.1		C
11-1.2		C
11-1.3		C
11-2		C
11-3		B
11-4		A
11-5		B
11-6		A
11-7		C
11-7.1		C
11-8		C
11-9		A
11-10		C
11-11		A
11-12		C
11-13		A
11-14		A
11-15		A
11-16		C
11-17		A
11-18		A
11-19		B
11-20		B
11-21		C

†SFC Software Feature Change

LIST OF EFFECTIVE PAGES (Continued)

PAGE	SFC †	REV
11-22		A
11-23		A
11-24		A
11-25		B
11-26		B
11-27		A
11-28		A
11-29		C
11-30		A
11-31		C
11-32		C
11-33		C
11-34		C
11-35		C
11-36		C
11-37		C
11-38		C
11-39		C
12-1		C
12-2		A
12-3		C
12-4		A
12-5		A
12-6		B
12-7		B
12-8		A
12-9		A
12-10		B
12-11		A
12-12		B
12-13		B
12-14		A
12-15		A
12-16		B
12-17		A
12-18		B
12-19		A
12-20		B
12-21		B
12-22		C
12-23		A

PAGE	SFC †	REV
12-23.1		C
12-23.2		C
12-23.3		C
12-23.4		C
12-23.5		C
12-24		C
12-25		B
12-26		B
12-27		B
12-28		A
12-29		A
12-30		A
12-31		B
12-32		B
12-33		A
12-34		B
12-35		B
13-1		C
13-2		B
13-3		C
13-4		B
13-5		B
13-6		C
13-6.1		C
13-7		B
13-8		C
13-9		C
13-9.1		C
13-9.2		C
13-9.3		C
13-10		C
13-11		C
13-11.1		C
13-12		B
13-13		C
13-14		B
13-15		B
13-16		C
13-17		C
13-17.1		C
13-18		C

PAGE	SFC †	REV
13-18.1		C
13-19		B
13-20		B
13-21		C
13-22		C
13-23		B
13-24		C
13-25		C
13-25.1		C
13-26		C
13-27		B
13-28		B
13-29		C
13-30		B
13-31		B
13-32		B
13-33		B
13-34		B
13-35		C
13-36		B
13-37		B
13-38		C
13-38.1		C
13-39		B
13-40		B
13-41		B
13-42		B
13-43		B
13-44		B
13-45		B
13-46		B
13-47		B
13-48		B
13-49		B
13-50		B
13-51		B
13-52		B
13-53		B
13-54		B
13-55		B
13-56		C

PAGE	SFC †	REV
13-57		C
13-58		C
13-59		C
14-1		C
14-2		A
14-3		A
14-4		A
14-5		A
14-6		A
14-7		A
14-8		C
14-9		A
14-10		A
14-11		A
14-12		B
14-13		A
14-14		A
14-15		A
14-16		A
14-17		A
14-18		A
14-19		A
14-20		A
14-21		A
14-22		A
14-23		A
14-24		A
14-25		A
14-26		C
14-27		A
14-28		A
14-29		A
14-30		A
14-31		A
14-32		A
14-33		A
14-34		A
14-35		A
14-36		A
14-37		A
14-38		A

†SFC Software Feature Change

LIST OF EFFECTIVE PAGES (Continued)

PAGE	SFC †	REV
15-1		A
15-2		A
15-3		A
15-4		A
15-5		A
15-6		A
15-7		A
15-8		A
15-9		A
15-10		A
15-11		A
15-12		A
15-13		A
15-14		A
15-15		A
15-16		A
15-17		A
15-18		A
15-19		A
15-20		A
15-21		A
15-22		A
15-23		A
15-24		A
15-25		A
16-1		A
16-2		B
16-3		A
16-4		B
16-5		A
16-6		A
16-7		A
16-8		B
16-9		B
16-10		A
16-11		A
16-12		A
16-13		C
16-14		A
16-15		A
16-16		A

PAGE	SFC †	REV
16-17		A
16-18		A
16-19		B
16-20		A
16-21		B
16-22		A
16-23		C
16-24		A
16-25		A
16-26		A
16-27		A
16-28		A
16-29		A
16-30		A
17-1		A
17-2		B
17-3		B
17-4		C
17-5		C
17-6		C
17-7		A
17-8		C
17-9		C
17-10		C
17-11		B
17-12		C
17-13		B
17-14		A
17-15		A
17-16		A
17-17		A
17-18		A
17-19		A
17-20		C
17-21		C
17-22		C
17-23		C
17-24		C
17-25		A
17-26		A
17-27		A

PAGE	SFC †	REV
17-28		C
17-29		C
17-30		C
17-31		C
17-32		C
17-33		C
18-1		C
18-2		C
18-3		B
18-4		B
18-5		B
18-6		B
18-7		B
18-8		B
18-9		B
18-10		B
18-11		B
18-12		B
18-13		B
18-14		B
18-15		B
18-16		B
18-17		B
18-18		B
18-19		B
18-20		B
18-21		B
19-1		A
19-2		C
19-3		A
19-4		A
19-5		C
19-6		A
19-7		A
19-8		A
19-9		A
19-10		A
19-11		C
19-12		C
19-13		C
19-14		C

PAGE	SFC †	REV
20-1		B
20-2		A
20-3		C
20-4		B
20-5		A
20-6		C
20-7		C
20-8		C
20-9		C
20-10		C
21-1		A
21-2		A
21-3		A
21-4		A
22-1		A
22-2		A
22-3		A
22-4		B
22-5		A
22-6		B
22-7		B
22-8		B
22-9		B
22-10		A
22-11		A
22-12		A
22-13		B
22-14		A
22-15		B
22-16		A
22-17		A
23-1		B
23-2		A
23-3		B
23-4		B
23-5		B
23-6		B
23-7		B
23-8		B
23-9		B
23-10		B

†SFC Software Feature Change

PAGE	SFC †	REV
23-11		C
23-12		B
23-13		B
23-14		B
23-15		B
23-16		C
23-17		C
23-18		C
23-19		B
23-20		B
23-21		B
23-22		B
23-23		B
23-24		B
23-25		B
23-26		B
23-27		B
23-28		B
23-29		B
23-30		B
23-31		B
23-32		B
23-33		A
24-1		C
24-1.1		C
24-2		C
24-2.1		C
24-2.2		C
24-3		C
24-3.1		C
24-3.2		C
24-4		C
24-4.1		C
24-5		C
24-6		C
24-7		C
24-8		B
24-9		C
24-10		C
24-11		C
24-12		C

PAGE	SFC †	REV
25-1		B
25-2		B
25-3		A
25-4		A
25-5		B
25-6		A
25-7		A
25-8		B
25-9		A
25-10		A
25-11		A
26-1		A
26-2		A
26-3		A
26-4		A
26-5		A
26-6		A
26-7		A
26-8		A
26-9		A
26-10		A
26-11		A
26-12		A
26-13		A
26-14		B
26-15		A
26-16		A
26-17		A
26-18		B
26-19		B
26-20		B
26-21		B
26-22		B
26-23		B
26-24		B
27-1		A
27-2		A
27-3		A
27-4		A
27-5		A
27-6		A

PAGE	SFC †	REV
27-7		A
27-8		A
27-9		A
27-10		A
27-11		A
27-12		A
27-13		A
27-14		A
27-15		A
27-16		A
27-17		A
27-18		A
27-19		A
27-20		A
27-21		A
27-22		A
27-23		A
27-24		A
27-25		A
28-1		A
28-2		C
28-3		A
28-4		B
28-5		A
28-6		A
28-7		A
28-8		A
28-9		A
28-10		A
28-11		A
28-12		A
28-13		A
28-14		B
28-15		A
28-16		A
28-17		A
28-18		A
28-19		C
28-20		C
28-21		C
28-22		C

PAGE	SFC †	REV
28-23		C
28-24		C
28-25		B
28-26		B
28-27		B
28-28		C
28-29		B
28-30		A
28-31		C
28-32		A
28-33		A
28-34		B
28-35		A
28-36		B
28-37		A
28-38		B
28-39		A
28-40		A
28-41		A
28-42		B
28-43		A
28-44		A
Index 1		A
Index 2		B
Index 3		B
Index 4		A
A-1 thru A-3		D
B-1 thru B-16		D
C-1 thru C-38		D
Cmt Sheet		-
Return Env		-

†SFC Software Feature Change

PREFACE

The purpose of this KRONOS 2.1 Workshop Manual is to provide the system analyst with detailed internal documentation readily available in a single manual. However, before the information presented herein is of value as reference material, the analyst should attend the KRONOS 2.1 Workshop. This workshop provides additional explanations for those areas which are not self-explanatory.

To accomplish the above objective, the manual provides detailed descriptions of system routines, including system interfaces, tables, and flowcharts. Some user interfaces are mentioned, however, all interfaces are described in other existing KRONOS 2.1 Manuals.

Participants in the workshop should be familiar with KRONOS 2.1 Time-Sharing and Usage, CP and PP COMPASS, and Operating Systems Theory.

Reference materials required during the workshop include current listings and reference manuals.

Current Listings

1) Catalog of a KRONOS 2.1 system
2) Dump of CMR. (This may be obtained by using the absolute dump program listed in Section 27.)
3) SYSTEXT (PPCOM and CPCOM)

Reference Manuals

1)	KRONOS 2.1 Installation Handbook	60407500
2)	KRONOS 2.1 Operator's Guide	60407700
3)	MODIFY Reference Manual	60281700
4)	KRONOS 2.1 Reference Manual	60407000
5)	KRONOS 2.1 Instant Manual	60407200
6)	KRONOS 2.1 Transaction Subsystem	60407900
7)	COMPASS Reference Manual	60360900

The following two charts summarize all the Tape handling control cards/macros. More information is available in the Reference Manual, Section 5.

	Change Number of Assignments	Change Number of Demands	Explicitly Associate Device & LFN	Write Label	Write VSN	Check For Label Match	Automatic Assignment	Requires Operator Assistance	Requires Special Permission
ASSIGN	Yes	No	Yes	No	No	No	Yes *1	Yes *2	Yes
BLANK	Yes & No *3	No	No	Yes *4	Yes	No	No	Yes	No
LABEL	Yes	No	No	Yes	No	Yes *5	Yes *1	Yes *2	No
REQUEST	Yes	No	No	No	No	No	Yes *1	Yes *2	No
RESOURCE	No	Yes	No	No	No	No	No	No	No
RETURN	Yes	Yes *6	No	No	No	No	No	No	No
UNLOAD	Yes	No	No	No	No	No	No	No	No

*1 Yes, if VSN specified on VSN card or VSN parameter on control card.

*2 Yes, if VSN not specified or duplicate VSNs are loaded on tape drives.

*3 If U parameter specified tape will not be unloaded, otherwise it will be.

*4 Label is only VOL1, HDR1.

*5 1. If R parameter specified, NOS checks tape labels against values on LABEL card, if comparison fails the job is aborted.

2. If W parameter specified, NOS checks tape labels against VSN on card, then writes labels from parameters on LABEL card. (See P. 5-93 Ref. Manual.)

3. If the lfn previously exists when the LABEL or REQUEST card is encountered, the LABEL or REQUEST card is treated as a NOOP.

*6 If Demand count = assign count reduce Demand count by one. If Demand count < assign count, do not change Demand count.

	Standard System	Level Support	Can Specify Block Size	Default Block Size	EOR	EOF	EOI	Write Terminating Condition
I	K2.1	0,17	Only in FET	1000B	Short PRU *1	Empty PRU (level 17)	*2	-
SI		0-17	Only in FET	1000B	Short PRU *1	Empty PRU (level 17)	*2	-
X	K2.0	No	Only in FET	1000B	Short PRU *1	Tape Mark	None	-
S	GEM Honey-well GE	No	Only in FET	1000B	Every PRU	None	*2	-
L	OEM (IBM)	No	Only in FET	Buffer Size	Every PRU	None	*2	-
E	-	-	FC	136	Tape Mark	Tape Mark	*2	Zero byte in byte 4
B	-	-	FC	150	Tape Mark	Tape Mark	*2	Zero byte anywhere
F	non-Cyber	-	FC	Must be specified (buffer size)	None	Tape Mark	None	-

*1 PRU = 128 words for coded
512 words for binary

*2 for labeled tapes: TAPEMARK, EOF1
for non-labeled tapes: None

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

FIGURES

FIGURES (Continued)

FIGURES (Continued)

FIGURES (Continued)

FIGURES (Continued)

FIGURES (Continued)

FIGURES (Continued)

FIGURES (Continued)

TABLES

TABLES (Continued)

KRONOS 2.1 INTRODUCTION 1

1.0 SCOPE

The KRONOS operating system controls the use of Control Data CYBER 70/Models 72, 73 and 74, and 6000 Series computer systems. Therefore, KRONOS is in control of the computer. KRONOS accepts input in the form of jobs submitted by users and processes them as directed by control cards accompanying each job as well as by keyboard commands from the console operator. The KRONOS operating system accepts jobs in four ways: time-sharing, local batch, remote batch, and system console input (Figure 1-1).

Efficient processing of user's jobs is the prime objective of the operating system. This section describes the inherent hardware characteristics, the basic software elements, and how they work together to accomplish the prime objective. Figure 1-2 shows the KRONOS system equipment configuration.

1.1 HARDWARE CHARACTERISTICS

KRONOS uses Peripheral Processor Units (PPU) for system and input/output tasks and a Central Processor(s) Unit (CPU) to execute user and system jobs. Central Memory (CM) contains user programs; system software areas are located at the lower end of Central Memory, Extended Core Storage (ECS) may be used by KRONOS.

1.1.1 Central Processor Unit

The CPU is designed to perform tasks of a computational nature; it has no input/output capability. It communicates with other system components through the central memory. Under KRONOS, the CPU is used almost exclusively for program compilations, assemblies, and executions. The CPU makes system requests through a CPU request register located at the Reference Address plus one (RA+1) of the current program in execution. However, that system work which can be done better in the CPU, is also processed in the CPU.

1.1.2 Peripheral Processor Units

The peripheral processor units from 1 to 20 (identified as PP0, PP1, . . . PPn) are identical; they perform many tasks for requesting programs in central memory. Peripheral processor unit(s) commonly referred to as PP(s) shall be so identified throughout this document. Each PP consists of PP memory of 4K, 12-bit, 1-byte words.

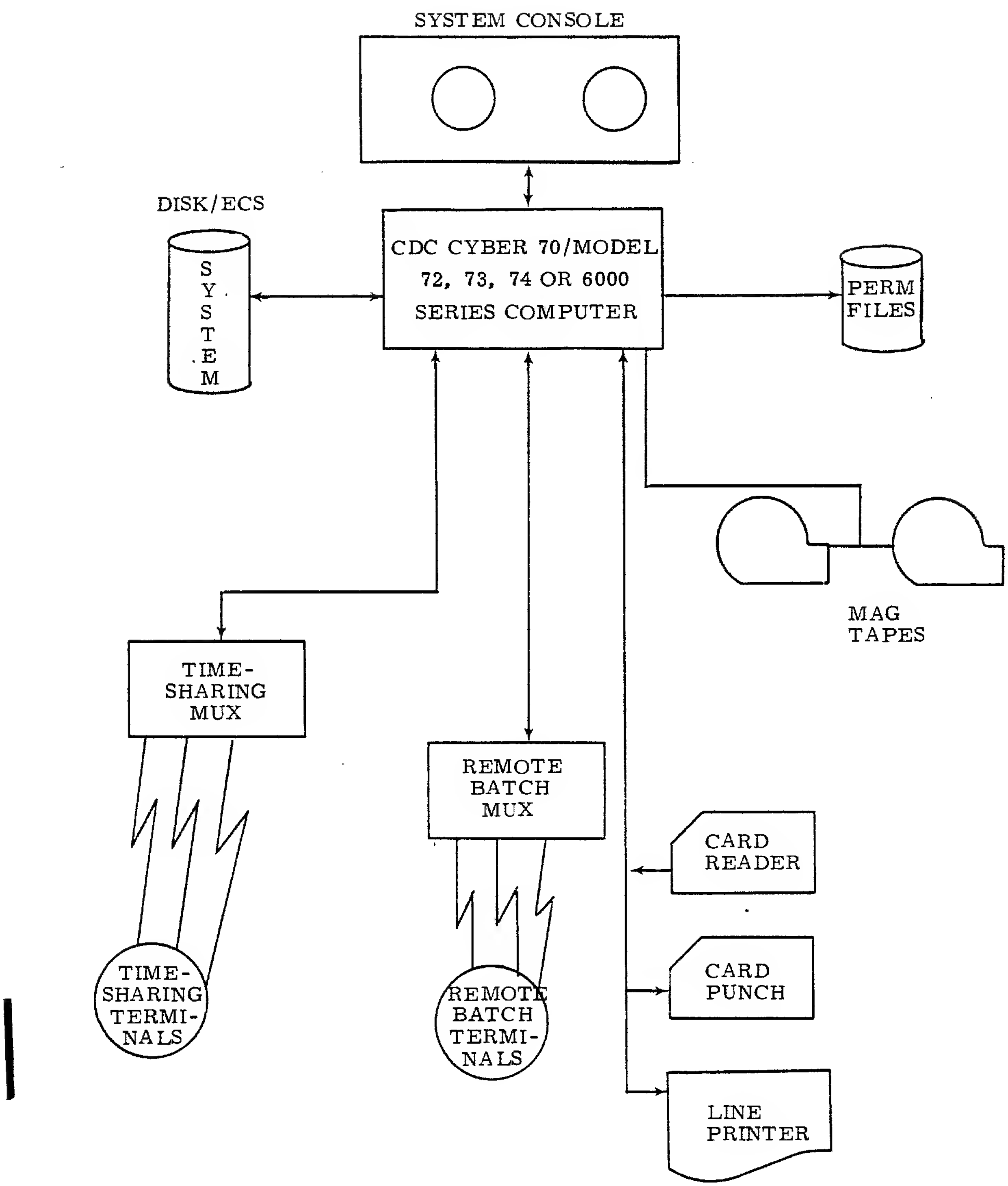
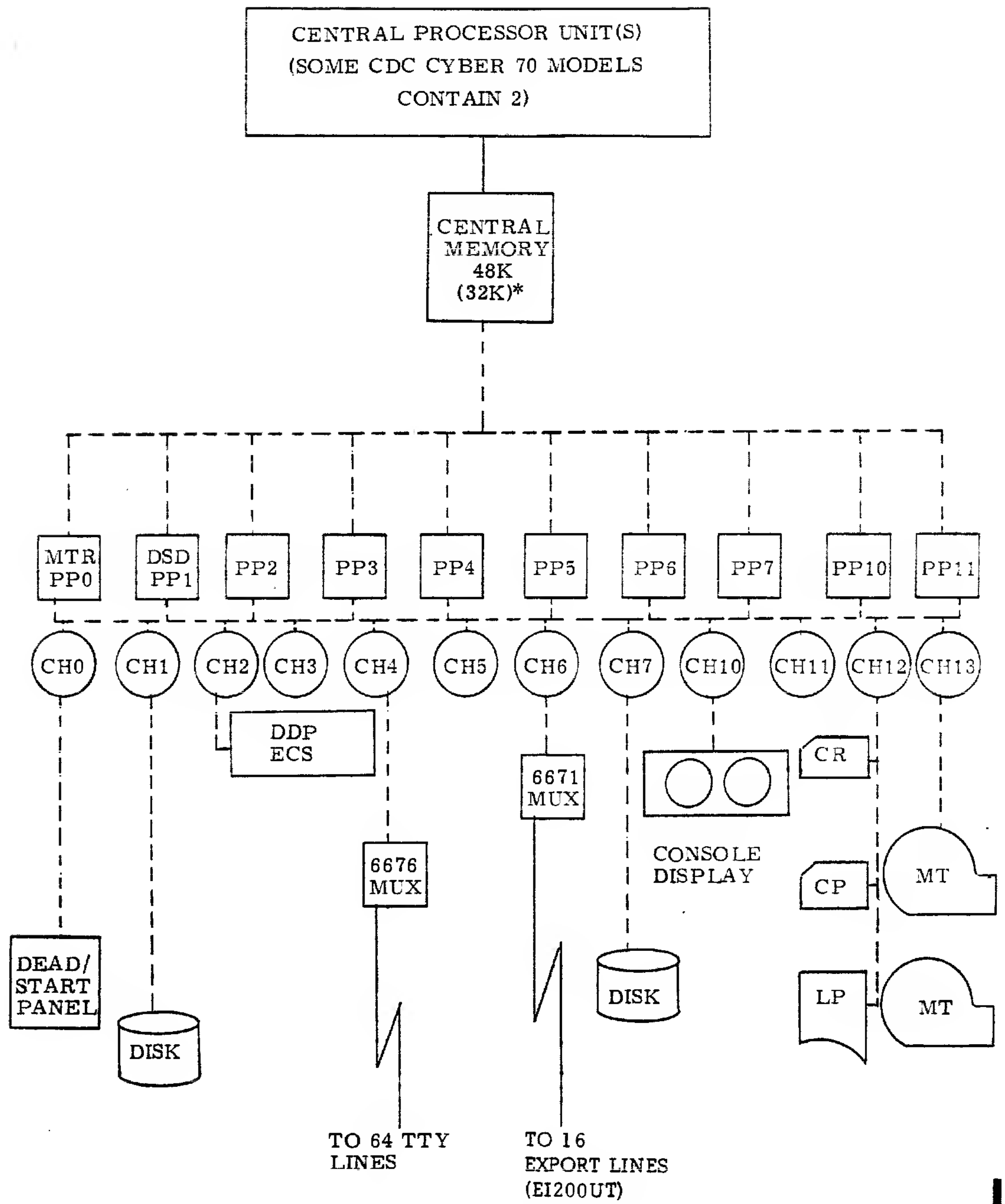


Figure 1-1. KRONOS System Equipment Configuration



*Due care at D/S for NCP, FNT size etc., needed for KRONOS 2.1 to operate in a system with 32K memory.

Figure 1-2. Typical Hardware Configuration

A PP can be assigned to control input/output, job scheduling, control card interpreting, system housekeeping and other tasks as required. Tasks are assigned one at a time to each PP by the CPU system monitor (CPUMTR). When an assigned task is completed, the PP signals the system. CPUMTR waits for this signal before assigning another task to the PP.

Each PP is assigned a block of eight words in the system area of central memory through which communications with the system are conducted. Each block contains an input register, an output register, and a message buffer.

1.1.3 Central Memory

Central memory words are 60 bits long; each is composed of five 12-bit bytes. Each 12-bit byte in a CM word is numbered 0 through 4, from left to right:

59 48	47 36	35 24	23 12	11 0
byte 0	byte 1	byte 2	byte 3	byte 4

One or more user programs may be in some state of execution concurrently under KRONOS. These programs are stored in central memory in an assigned user area; a set of system components necessary for the operation of the system is also stored in central memory, forming Central Memory Resident (CMR). Central memory is accessible by all PP's and CPU(s) and forms the communications link between all processor units in the computer system.

CMR contains system communications areas, system tables, CPU resident routines, the library directory, and information about each job currently in execution.

1.2 EXTENDED CORE STORAGE

Under KRONOS 2.1, Extended Core Storage (ECS) may be used as a high-speed peripheral storage device via a TRT (Track Reservation Table).

1.3 SOFTWARE ELEMENTS

Two elements are basic to the KRONOS operating system: files and control points.

1.3.1 Files

A file is an organized collection of data known to the system by a given name. Data is organized in one or more logical records and terminated by an End-of-Information (EOI)

indicator. Under the KRONOS operating system, the jobs it processes and all intermediate and final results are contained in files or parts of files.

1.3.2 Control Points

The system can control execution of several jobs at one time. When placed into CM before execution, each job is assigned a value which is the control point number and the index to a control point. Jobs at control points are assigned to a processor for execution. Each control point area in CMR holds all information necessary to process the assigned job.

1.4 KRONOS ORGANIZATION

The KRONOS operating system consists of PP programs, CPU programs, macro definitions and symbol definitions. The entire system is contained in a magnetic tape file produced by the library maintenance program MODIFY. Programs in the library file are in source language form. Installation options are provided to permit flexible selection of system features during the assembly and creation of a deadstart file on tape.

A system monitor is in complete supervisory control of the hardware system. The system monitor is made up of PP overlay MTR which operates in PP0, and CPUMTR which is assembled as part of the central memory resident (CMR).

1.5 CENTRAL PROCESSOR AND KRONOS

1.5.1 CM Organization

The allocation of central memory is illustrated in Figure 1-3.

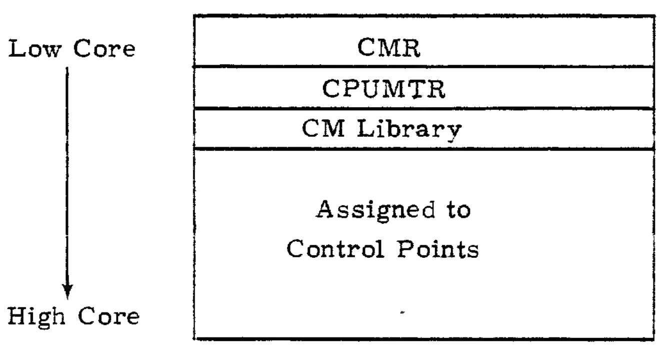


Figure 1-3. Central Memory Organization

Low core is allocated to the central memory resident portion of KRONOS and executable system programs. The remaining area can be assigned to control points.

1.5.2 Control Point Concept

Blocks of central memory storage not allocated for system use are ordered by control point number and assigned to jobs. Each control point number has a corresponding table in CMR called the control point area. A control point is not a physical entity, but rather a concept used to facilitate bookkeeping. The control point number and the control point area, however, are physical quantities that do appear in the system.

Under KRONOS 2.1 any number of control points, up to 23D (decimal) are possible. In the released system, the default value is 23 decimal. In an installation with n control points for user jobs they are numbered from 1 to n. A job assigned to a control point is identified by its control point number; only one job can be assigned to a control point at any one time. Once a job is assigned to a control point, system resources such as central memory, channels, equipments and processors may be assigned to the control point for use by the job.

Storage assigned to a single control point is contiguous; storage for all control points is not necessarily contiguous. The core storage block assigned to the job at control point 2 is higher than the block for the job at control point 1, and storage for control point 3 is always higher than that for control point 2, and so on.

In Figure 1-4, no storage is assigned to control points 3 and 5; unassigned storage appears between assigned storage.

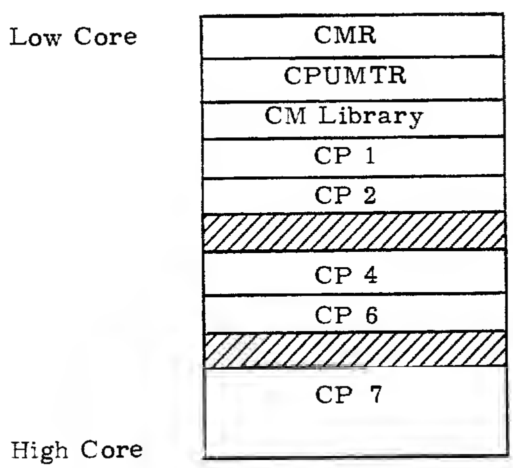


Figure 1-4. Control Points in Central Memory

1.5.3 Sub Control Points

Sub Control Points (SCP) is another concept of Control Data software. A SCP runs exactly the same way as a Control Point (CP) except that it is contained within one CP. The user writes a controlling segment, an Executive (much like a user-oriented CPUMTR), which via CPUMTR RA+1 type requests can control a number of SCPs at its CP. The major use for this CP tasking is to protect the controlling segment from any of its subroutines.

1.5.4 Special Control Points

In addition to the n control points used for running jobs, control point n+1 and a pseudo control point numbered zero are used by the system.

Control point zero is used to identify system resources not allocated to a job at a control point; they are deallocated or allocated to the system. If an equipment is assigned to a control point, that number is entered into the system table entry for that equipment.

If not assigned to a job, the equipment is assigned to control point zero and is available to be assigned to a job. All active system files are attached to control point zero. They include the system file, any job files that have been read in and are waiting for scheduling, and all output files waiting to be processed by BATCHIO. Control point n+1 is used by CPUMTR to process certain CPUMTR functions. Any CPUMTR function which uses more than some preset amount of CPU time is assigned to the system control point n + 1 by CPUMTR. The system control point is treated as any other control point by the system with the exception that its queue priority is so high, it can never be rolled out. Its CPU priority is the highest in the system (100).

1.5.5 Job Description Number (FNT queues ordinal)

During the course of execution, a job might not remain continuously at the same control point. It is possible for the job to be rolled out while it is only partially executed. When a job is rolled out, it is not associated with a control point. When a job is rolled back in, it is probably associated with a control point other than the control point during its original assignment.

During the time a job is rolled out, the only table in CMR that contains information about the job is the FNT with type rollout.

1.5.6 Storage Moves

Since jobs come and go as they finish processing and new jobs begin, or as jobs are rolled in and out, CM storage must be reallocated and jobs must be moved. If a job at a control point requests additional storage, it may be necessary to move jobs to obtain the required storage.

A request for a reduced field length merely resets the FL (field length) size in the control point area; no storage move takes place. A request for an increased field length, when the total already associated with the control point is adequate, will result in resetting the FL size in the control point area; no storage move will take place.

If it is necessary to take unallocated storage from other control points to satisfy a request for increased field length, control points above and below the requesting control point will be scanned. This scan locates the combination of unallocated storage blocks which will result in a move of the least amount of storage.

In Figure 1-4 shown under Control Point Concept, if control point 1, needs more storage, it will be necessary to move control point 2. If control point 6 needs storage, sufficient unallocated storage may be available to make a control point move unnecessary. If, however, control point 7 needs additional storage, control points 4, 6 and 7 will be moved downward to provide the storage. Added storage always extends the field length upward.

1.5.7 CP System Communication

A running CP program must communicate with the system as described in the following examples:

1. When a CP program is loaded and executed as a result of a control card call, the system must place any parameters specified on the control card in an area where they can be read by the CP program.

2. No CP instructions allow a CP program to perform input/output; therefore a CP must send a request to the system, to load a PP program to execute the input/output.

3. When a CP program terminates, it must advise the system that it may process the next control card.

Since a CP program cannot access memory locations outside its field length, any area reserved for communication between a CP program and the system must be within the field length of the job. The first 100B(octal) locations of each job's field length are reserved for this purpose. The first program loaded into a user field length is always loaded at location RA+100 (for the user, this is location 100). The RA area is shown in CMR Section 2.

1.5.8 CP PP Communication

If a CP program wishes to call a PP program it places the PP programs name and up to two arguments in RA+1. If Auto-recall is desired bit 40 is set. If the Central Exchange Jump (CEJ) installation is available, the user's program should use it immediately after placing a call in RA+1. This will cause CPUMTR to begin execution immediately. If CPUMTR determines that the RA+1 call should be assigned to a PP, CPUMTR will write the RA+1 word into the PP's input register in CMR. The name and any parameters in bits zero through 35 appear in the input register exactly as they did in RA+1. Parameters are passed from a CP program to a PP program through this parameter field. The format for the PP communication area is shown in CMR Section 2.

For example, if the PP program CIO is called, CIO will find the relative address of the File Environment Table (FET) to be used in the operation by reading its input register. It can find the RA of the control point field length by reading the control point number from its input register, computing the address of the control point area, and reading the value of RA from the control point area. By adding the RA to the relative FET address, CIO obtains the absolute address of the start of the FET. CIO then reads the parameters for the I/O operation from the FET.

MTR continually scans RA+1, in the event that the users program does not use the central exchange jump, or the instruction is not available. When a RA+1 call is found, MTR initiates CPUMTR. Less CPU time is used by letting CPUMTR process the call, than if MTR did it directly.

1.5.9 Program Recall

The recall program status is provided in KRONOS to enable efficient use of the central processor and to capitalize on the multiprogramming capability of KRONOS. Often, a CP program must wait for an I/O operation to be completed before more computation can be performed. To eliminate the CPU time wasted if the CP program were placed in a loop to await I/O completion, a CP program can ask KRONOS to put the control point into recall status until a later time; the CPU may be assigned to execute a program at some other control point.

Recall may be automatic or periodic. Auto-recall should be used when a program requests I/O or other system action and cannot proceed until the request is completed. KRONOS will not return control until the specific request has been satisfied. Periodic recall can be used when the program is waiting for any one of several requests to be completed. The program will be activated periodically, so that it can determine which request has been satisfied and whether or not it can proceed.

1.5.10 Periodic Recall

To enter periodic recall, a CP program puts the characters RCL left-justified into RA+1. On encountering the RCL request, the system assigns the CPU to some other control point. After a certain interval of time has elapsed, the control point is restarted and the CPU is again assigned to execute the program at the control point. At this time, the CP program can check the completion bit in the FET to see if the I/O is finished. If so, the CP program may proceed with computations. If I/O is not complete, the CP program can put itself back into recall.

1.5.11 Automatic Recall

If a CP program makes a request in RA+1 and bit 40 of RA+1 is set to one, the control point will be put into automatic recall after the request has been initiated. Again, the CPU is assigned to another control point as in periodic recall. In this case, however, the program in recall will be restarted by CPUMTR after the PP has dropped or issued the RCPM functions. The completion bit in the FET is never statused. The only criterion for CP start-up is the RCPM or drop.

Recall and auto-recall are most often used while waiting for CIO to process an I/O request; however, any time a PP program is called from RA+1, with bit 40 of RA+1 set to one, the control point will be put into auto-recall. If bit 40 is set, bits zero through 17 of RA+1 must contain the address of a word in the program's field length called a reply word. When the PP has completed its function, it will set the completion bit (low order bit) in the reply word, and drop or RCPM. The completion has no basic significance to NOS.

For a call to CIO, the reply word is the first word of an FET. For other programs the reply word need not be part of an FET.

A CP program can put itself into auto-recall without calling a PP program by putting RCL left-justified in RA+1 and setting bit 40 of RA+1 to one. Bits zero through 17 of RA+1 must contain the address of a reply word. A program which has already initiated one or

more I/O operations might go into auto-recall in this way, using the first word of the FET associated with one of the I/O operations as the reply word. Figure 1-5 shows the formats of RA+1 for: a normal CIO call; a request for periodic auto-recall; a CIO call with auto-recall bit set; and an RCL call with auto-recall bit set. For periodic recall, a user must issue a normal CIO call followed by an RCL request. For auto-recall, only one request is required.

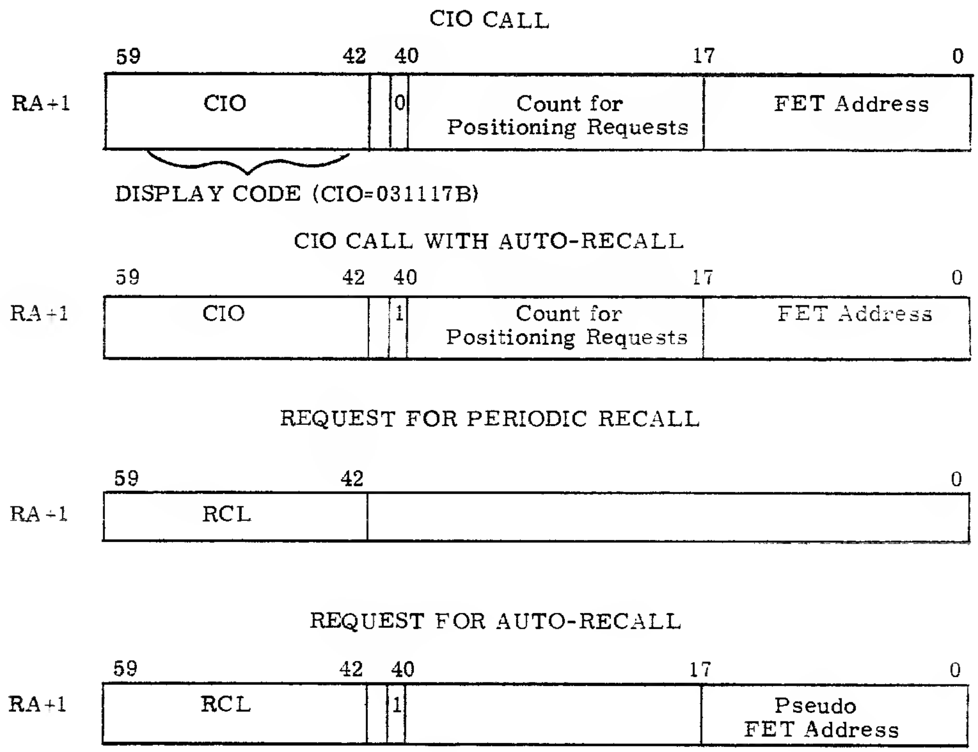


Figure 1-5. RA+1 CIO and Requests Calls

SPECIAL NOTE TO PP PROGRAMMERS

Any CP program making a call to a PP program using auto-recall needs to be restarted by the PP program unless the PP program intends to drop before the CP program is started up. Just setting the completion bit in the pseudo FET word is not enough to get the CP program restarted. In addition, the PP routine must issue the monitor function RCPM, request CPU, to get the CP program restarted. See Section 23 routine TLP for an example of the above. Also note that unless a CP program has QP ≥ MXPS, all call to PP programs with the exception of CIO will be forced into auto-recall by CPUMTR.

RA+1

I. With FET address specified

RA+1

XXX	R		fet

conditions for startup of C.P.

1. a. RCPM or

   b. DPPM

II. System forced auto-recall without FET address

XXX	0		0

1. a. RCPM or

   b. DPPM

Thus a user can be in Auto-Recall without PP activity.

Remember all calls are put into Auto-Recall automatically-except CIO, or RCL.

For the user there is no difference between user's set and system forced onto recall.

Auto-recall initiated by the RECALL macro is treated as follows: CPU monitor checks the completion bit and if set takes the CP out of auto-recall. If not set, CPU monitor leaves the RCLP in RA+1 and exits. This request will be detected later by MTR, who will call CPUMTR.

Normally, CP programs use auto-recall for convenience, but only one request involving auot-recall can be processed at one time. For example, to initiate I/O action on several files at once, a user must employ the periodic recall tecnique. He will issue all the requests without recall (using a separate FET for each request); then go into periodic recall. Each time the CP program is restarted by the system, it can check all the files for completion and go back into periodic recall if any are still incomplete.

Periodic recall may be used also when a CP program can initiate an I/O request and then perform some computation. In some cases, the I/O would be completed before the computation; in others, the computation would be done first. The user would go into recall only when computation was done, and then only if the I/O was still in process.

Periodic recall should also be used, if possible, to continue processing while only part of the data buffer has been read or written by the I/O driver.

During normal operation central memory queue priority and CPU priorities are as shown in Figure 1-6. KRONOS supports 23D Control Points. Queue priority governs which jobs in the Input Queue gain access to CPs. CPU priority governs which jobs at CPs gain access to the CPU.

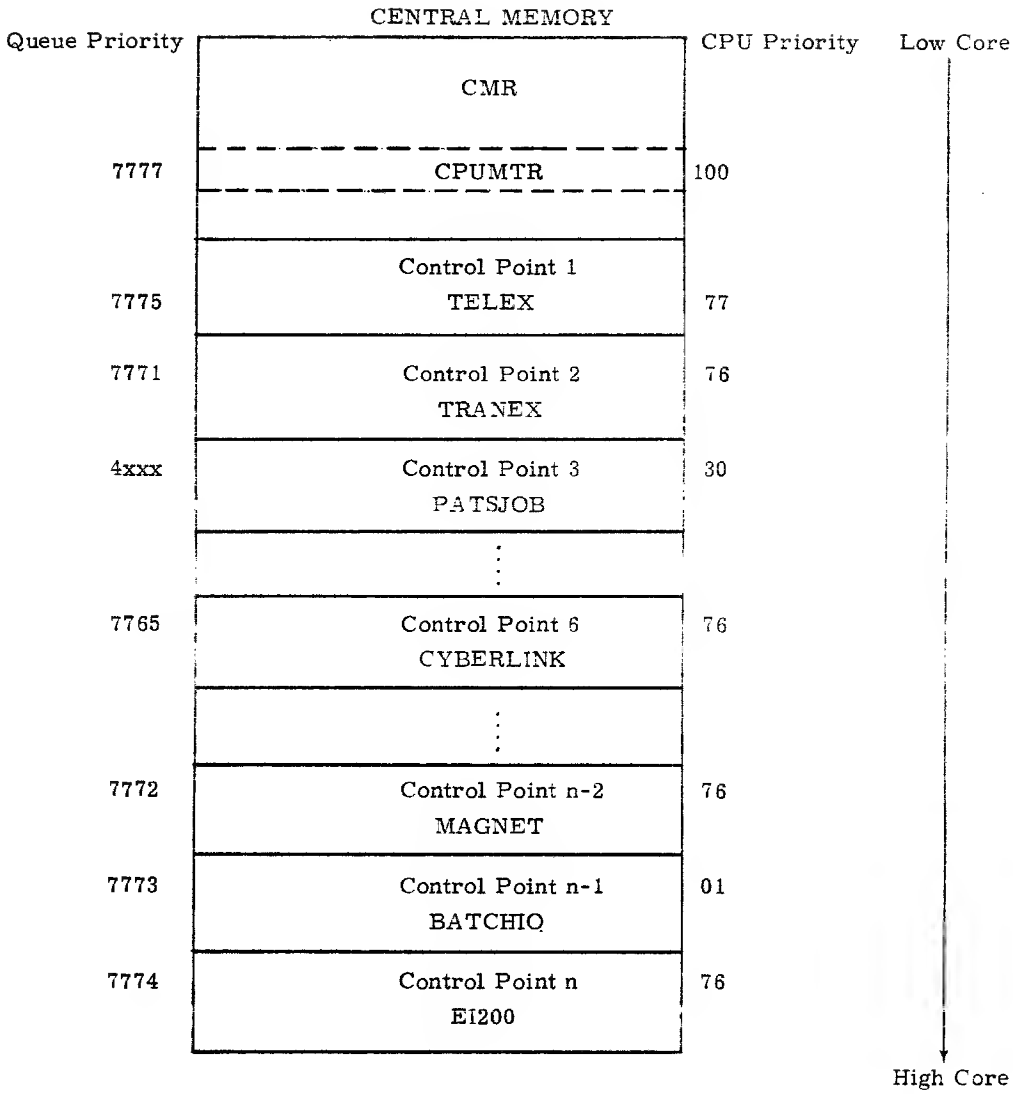


Figure 1-6. Central Memory Priorities

The following definitions are used extensively in KRONOS 2.1. A graph of CPU or CM time slice is provided to graphically point out the difference between these two very basic concepts.

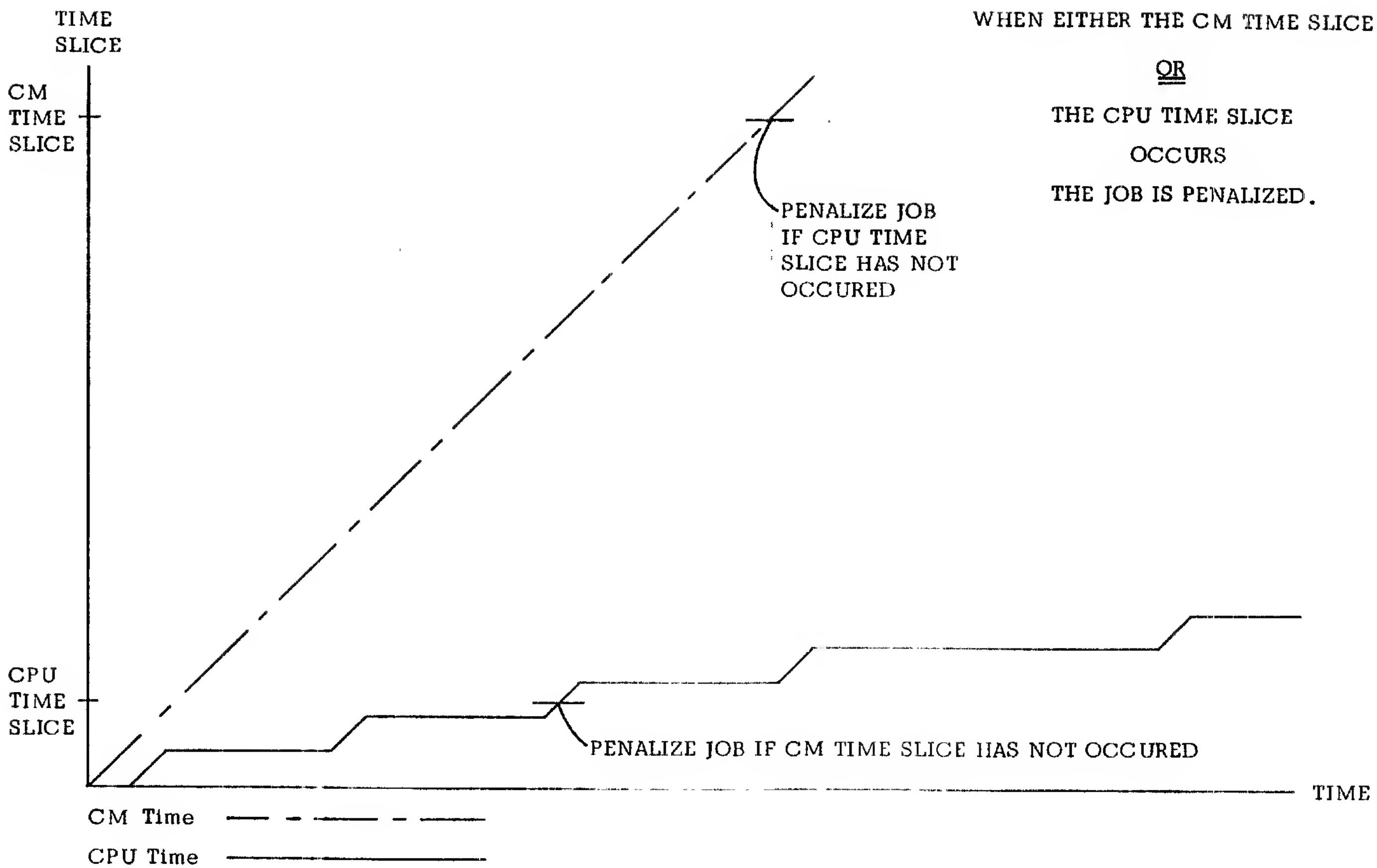
TABLE 1-1. SYSTEM RESOURCE TIMES

ITEM	DESCRIPTION
Queue priority	The priority which governs entry into a CP from the INPUT or ROLLOUT queue and also governs disposition to a printer.
CPU priority	The priority which governs which candidate for the CPU will get access to the CPU.
CPU time slot	That time period when the CPU is shifted from one candidate to another.
CPU time slice	The total time period that a CP can use the CPU without being penalized.*
CM time slice	The total time period a CP can reside in a CP without being penalized.*

*The queue priority in CPA is reduced to the LQP for this orign type.

TABLE 1-2. JOB ORIGINS

SOURCE	END
SYOT	SYSTEM
BCOT	BATCH
EIOT	EXPORT
TXOT	TELEX
MTOT	MULTI-TERMINAL



The CM time - increases linearly with time as long as the job is at a CP without respect to the use of the CPU.

The CPU time - increases as a step function with a linear ratio only while the job is actually <u>using</u> the <u>CPU</u>.

Figure 1-7. Graph of CM Time Slice and CPU Time Slice

CENTRAL MEMORY RESIDENT 2

2.0 INTRODUCTION

The low end of central memory is reserved by the KRONOS operating system and provides the major coordinating area for system operation. Central Memory Resident (CMR) contains pointers, tables, central monitor (CPUMTR), libraries, and library directories. The length of CMR is dependent upon several factors, including the number of peripheral processors, the number of control points, the number of mass storage devices, and others. Figure 2-1 shows an overview of the layout of CMR giving the relative positions of the various parts of CMR. Subsequent discussions describe in more detail the many pointers and tables resident in CMR.

Address	Area	Description
0 77	SYSTEM POINTERS AND CONTROL WORDS	
100 105	CHANNEL STATUS TABLE	
106 177	MORE RESERVED SYSTEM POINTERS	
200	CONTROL POINT AREAS	200B words for each control point
(N+1) *200	SYSTEM CONTROL POINT AREA	
(N+2) *200	DAYFILE BUFFER POINTERS	3 entries for the three system dayfiles (SYSTEM, ERRLOG, ACCOUNT) + 1 entry for each control point; 2 words for each entry
	PPU COMMUNICATION AREA	10B words for each PPU
	(EST) EQUIPMENT STATUS TABLE	1 word entry for each type of equipment, 100 words total
	(FNT/FST) FILE NAME/ FILE STATUS TABLE	2 word entry for each file active in the system
	(MST) MASS STORAGE TABLES AND TRACK RESERVATION TABLES (TRT)	MST consists of MSTL (currently = 20B) words for each mass storage device. TRT immediately follows the MST for each device and is size dependent on device.
	JOB CONTROL AREA	
	DAYFILE BUFFERS	All buffers for all CPs and the 3 system dayfile buffers 100B words per buffer
	DAYFILE DUMP BUFFER	Used by (1DD) dayfile dump routine called from DFM when dumping dayfiles. Only one PPU can dump dayfiles at a time.
	ECS/PP BUFFER	Used to transfer ECS words to PP memory or other ECS locations. It is 100B words long to ensure that CM is not locked out and that the read pyramid is not tied up. In this way critical PP routines like 1TD, 1MT, DSD are not locked out of CM and miss data scans.
	CPUMTR	
	(RPL) RESIDENT PERIPHERAL LIBRARY	
	(RCL) RESIDENT CENTRAL LIBRARY	
	(PLD) PERIPHERAL LIBRARY DIRECTORY	
	(CLD) CENTRAL LIBRARY DIRECTORY	
	AVAILABLE CORE	

Figure 2-1. CMR Layout

TABLE 2-1. FNT/FST MASS STORAGE FILE TYPES

FILE TYPES	VALUE	DESCRIPTIONS
	Files in Queues	
INFT	0	INPUT
ROFT	1	ROLLOUT
PRFT	2	PRINT
PHFT	3	PUNCH
TEFT	4	TIMED/EVENT ROLLOUT
QUFT * 1	5	END OF FILE IN QUEUES
	Other Files	
SYFT	5	SYSTEM
LOFT	6	LOCAL
CMFT	7	COMMON
LIFT	10	LIBRARY
PTFT	11	PRIMARY TERMINAL
PMFT	12	DIRECT ACCESS PERMANENT FILE
FAFT	13	FAST ATTACH FILE
	Cyberlink Files	
HSFT	14	CYBERLINK TRANSMIT
LCFT	15	WAIT USER ACCESS FILE
CNFT	16	IN CYBERNET FILE
MXFT * 2	17	MAXIMUM NUMBER OF FILE TYPES

* 1 Used by the system as an upper limit for checking of types (i.e., all types less than QUFT are type QUEUE.

* 2 See * 1 (i.e., any types greater or equal MXFT are illegal).

TABLE 2-2. NON FNT/FST TYPES

TYPE	VALUE	DESCRIPTIONS
	Job Origin	
SYOT	0	SYSTEM
BCOT	1	BATCH
EIOT	2	E/I 200
TXOT	3	TELEX
MTOT	4	MULTI TERMINAL
MXOT * 1	5	MAXIMUM JOB ORIGIN TYPE
	Error Flags	
TLET	1	TIME LIMIT
ARET	2	ARITHMETIC ERROR
PPET	3	PPU ABORT
CPET	4	CPU ABORT
PCET	5	PP CALL ERROR
ODET	6	OPERATOR DROP
PSET	7	PROGRAM STOP
FLET	10	FILE LIMIT
TKET	11	TRACK LIMIT
SYET	12	SYSTEM ABORT
MXET * 2	13	MAXIMUM NUMBER OF ERROR FLAGS
	Pseudo Channels	CREATE NEW FNT ENTRY
FECT *3	14	FILE ENTRY PSEUDO CHANNEL
FNCT	15	FILE NAME TABLE UPDATE EXISTING FNT ENTRY
EBCT	16	ECS BUFFER

* 1 See * 1 p.2-1 (i.e., all job origin types must be less than MXOT).

* 2 See * 1 p.2-1 (i. e., error flags greater or equal MXET are illegal).

* 3 The two pseudo channels FECT and FNCT can be redescribed as follows:

FECT create new FNT entry. FNT entries are denoted as empty when the lfn equals zero. OBF, with the exception of 1TA for TELEX, is the only routine which creates new FNT entries. In order to protect from two OBFs attempting to create new files in the same FNT entry, this pseudo channel is used. When OBF decides where to create its entry, it requests this pseudo channel. Then it can enter the file with no fear of some other OBF simultaneously trying to use the same entry.

FNCT update existing files. Primarily used by CIO to update the status and position information in the FST.

The pseudo channel EBCT is used to interlock the ECS/PP buffer in CMR.

TABLE 2-3. FIXED SYSTEM PRIORITIES

NAME	VALUE	DESCRIPTIONS
MXPS *	7760	MAXIMUM PRIORITY FOR ROLLOUT
CLPS	7765	CYBERLINK
MSPS	7766	MASS STORAGE CONTROL
STPT	7767	TRANSACTION STIMULATOR
STPS	7770	TELEX STIMULATOR
TRPS	7771	TRANEX
MTPS	7772	MAGNET
BIPS	7773	BATCHIO
EIPS	7774	E/I 200
TXPS	7775	TELEX
MNPS	100	MINIMUM SCHEDUABLE
FRPS	1	FORCED ROLLOUT
ERPS	2	JOB ERROR
FEPS	3	FORCED ROLLOUT, NO FL
FFPS	4	FORCED ROLLOUT, WITH FL

* Priorities above MXPS are used by subsystems for identification as well as scheduler control.

The following is true for QP of MXPS:

1. QP ≤ MXPS: job can be rolled out and can be penalized for exceeding its time slices.
2. QP = MXPS: job can only be rolled out by subsystems and is <u>not</u> penalized for exceeding its time slices.
3. QP ≥ MXPS: job can <u>not</u> be rolled out, and is <u>not</u> penalized for exceeding its time slices.

Jobs which are rolled in by operator action or waiting for operator tape assignment are given QP = MXPS.

TABLE 2-4. POINTERS AND CONSTANTS – ALPHABETICAL LIST OF NAMES

NAME	VALUE	DESCRIPTIONS
	System Constants	
CHDS	10	Display Channel
CHMT	13	Magnetic Tape Channel
DFDS	230	Dayfile Dump buffer length
MPRS	100	Monitor function priority
NMSD	20	Maximum number of Mass Storage devices
NPFS	4	Number of P.F. activities allowed
NROS	2	Number of simultaneous Rollout/Rollin
	Pointers	
ACML	23	Available central memory
ACPL	60	Active CPU status
CLD	25	Central Library Directory
CLDP	7	CLD Pointer
CMCL	57	Central Memory control image (MTR)
CPUL	1	CPU configuration
CTIL	100	Channel Status Table
DFPP	3	Dayfile pointer
DTEL	31	Date (DSD)
ESTP	5	EST Pointer
FNTP	4	FNT Pointer
IPRL	42	Installation parameters
JBCP	4	Job Control Area pointer
JDAL	26	Julian date
JSCL	40	Job Scheduler Control
JSNL	22	Job Sequence Number
MFLL	1	Machine field length
MSAL	107	Mass Storage fill assignment (entry = 4000B + equipment number)
MSCL	24	Monitor scan control
NCPL	2	Number of Control Points
PDTL	27	Packed date
PFNL	110	P.F. activity control
PLD	24	FWA of Peripheral Library directory
PLDP	2	PLD pointer
PPAL	47	IR address of next available PPU
PPCP	2	PP Communication area pointer
PPUL	1	PP Configuration
PXPP	62	PP Exchange area pointer
PCLP	6	Resident CPU Library pointer
RPLP	1	Resident PPU Library pointer
RTCL	106	Real Time Clock image (MTR)
SFPL	64	*SFP* auto load code
SPLP	46	System PLD pointer
SSCL	44	Sub-System Control words (C.P. numbers)
SSTL	43	System Status
TIML	30	Time of Day (DSD)

TABLE 2-5. CONTROL POINT AREA – ALPHABETICAL LIST OF NAMES

NAME	N*200 +VALUE	DESCRIPTION
AACW	75	Account access control word
ACTW	50	Start of accounting words
ACTWE	54	End of accounting words
ACTWL	--	Length of accounting words
ACUW	74	Account Central memory usage
APJW	72	Account project number word
APUW	73	Account peripheral usage word
CMUM	51	Central memory usage
CPJW	54	Central processor time start of job (TXOT only)
CPNS	-	First control point number
CPTW	50	Central processor time
CSBW	130	Control Statement buffer
CSBWE	200	End of Control statement buffer
CSPW	67	Control statement pointer
CTLW	24	CPU time limit
DBAW	66	K and L Display control word
EECW	65	ENTRY/EXIT Control
FLCW	60	Field length control
ICAW	76	Inter-control point communication control
JCIW	22	Job Control information
JCRW	102	Job Control registers
JNMW	21	Job Name
LDCW	61	Loader Control word
MSUW	52	Mass storage usage
MS1W	30	Message 1 area
MS2W	35	Message 2 area
MTUW	53	Magnetic tape usage
OAEW	21	Operator assigned equipment
PFCW	63	Permanent file control word
RFCW	71	Resource file control word
RLPW	25	PP recall register
SEPW	100	Special Entry point Word
SNSW	26	Sense switches
SPCW	101	System processor call word
STSW	20	Status word
TIAN	62	Terminal interrupt address
TINW	65	Terminal input pointer
TIOW	62	Terminal I/O pointers
TSCW	23	Time Slice Control
UIDW	64	User identification
UPCW	77	User profile Control word

TABLE 2-6. REMAINING CENTRAL MEMORY - ALPHABETICAL LIST OF NAMES

NAME	OFFSET VALUE	DESCRIPTION
CLD	25	CLD - CPU Library directory Entry = 2 words.
DEVL	4	Device allocation information
DFBP	0	Dayfile buffer pointers. Entry = 2 words.
EST	0	Equipment status table. Entry = 1 word
ETB	1	System event tag
FNT	0	File name table. Entry = 2 words.
INQT	0	Input file control
ISTL	15	Installation use
JBC	41	Job control area
MSDL	1	Mass storage driver mods
MST	0	Mass storage table
MSTL		Length of MST
MXQT	3	Maximum number of queue control words
OTQT	2	Output file control
PFCT	5	Permanent file control
PFDL	6	Permanent file description
PFIL	5	Permanent file interlock word
PFUL	7	Permanent file user description
PLD	24	PPU library directory
PPC	202	PPU communication area
RCL	21	Resident CPU library
ROQT	1	Rollout file control
RPL	23	Resident PPU library
SVJT	3	Service control
TRT	20	TRT - Track Reservation Table. Entry = 1 byte ÷ 1 bit for each track.
TRTL	0	TRT definition

Word	59	47	35	29	23	11	
000			Zeros				
001	FWA Resident PP Library		Number of PPUs		*1 CPU Config.	Machine FL	RPLP, PPUL, CPUL, MFLL
002	FWA PP Library Directory				Number of CTRL PTS	PP Comm. Area Addr	PLDP, NCPL, PPCP
003	Dayfile PNTR FWA	FWA Dayfile DUMP Buffer				No. Exces *6 Dayfiles	DFPP
004	FWA FNT	LWA+1 FNT			FWA Job Control Area		FNTP, JBCP
005	FWA EST	LWA+1 EST	LWA+1 RMS Equipment		FWA ECS/PP Buffer		ESTP *5
006	FWA CPU Library						RCLP
007	FWA CPU Library Directory		FWA COS Format CPU Lib Directory				CLDP
010–017	Installation Area						
020						CMR Size/100 B	
021	System Name						
022				Job Sequence Number Counter			JSNL
023						Avail CM /100 B	ACML
024	Job Scheduler	CPU Recall	PP/Auto Recall		Job Advance	Job Switch	MSCL (Delay Word)
025	Reserved						
026				Julian Date			JDAL
027			Year	Mo	Day, Hrs	Min, Sec	PDTL
030	HH.MM.SS.						TIML
031	YY/MM/DD.						DTEL
032–037	System Date Line						
040	Bit 12 is Scheduler Requested Flag →					Scheduler Cycle	JSCL
041	← Bit 59 is Scheduler Active Flag				18 bits *3	18 bits *4	
042			Assumed Char. Set Conversn.		Assumed Conver. Mode	Assumed Tape Den	IPRL
043	*2 See Footnote						SSTL
044	Reserved	TELEX	E/I200		BATCHIO	MAGNET	SSCL (Subsystem Control Words)
045	TRANEX	TELEX Stim.	TRANEX Stim.		Reserved	CYBER-LINK	
046	Pointer to Non Alternate Device PLD		No. CNTL PTS			Addr. PP Comm. Area	SPLP
047						IR Addr. Next Avail PP	PPAL if 0, No PP is Available

* Descriptions follow:

Figure 2-2. Pointers, Constants, and Control Words

* 1	Bit 15	CMU Present
	14	CEJ/MEJ Option Present
	13	CPU 0=6600 Present
	12	CPU 1=6400 Present

* 2 Bit 0-Disable Autoroll, 1-Disable Job Scheduler, 2-Disable Priority Eval, 12-Debug Switch, 13-Console Init. Lock Status 42-Disable Removable Device Checking, 43-Disable Tranex, 44-Disable Magnet-45-Disable EI200, 46-Disable Telex, 47-Disable BATCHIO, 48-Disable Acct. Verification, 49-Ignore Acct. Card.

* 3 Delay for 1SJ to call 1SP

* 4 Delay for 1SP to call 1CK

* 5 Mass storage equipment can be mixed with non mass storage equipment in the 1st with the following restrictions:

  1. Eq 0 must be MS if defined.

  2. No MS may be specified beyond the CMR pointer for LWA + 1 of MS in word 5 (ESTP). (i.e., the operator may not dynamically set any MS devices after this pointer in CMR with memory entry commands.) The system builds this pointer when the EST is created at D/S time.

* 6 Number of dayfile, besides CP. So normally = 3 for 3 system dayfiles.

Figure 2-2. Pointers, Constants, and Control Words (Continued)

Used by CPU-MTR (060–105)

Address	59	47	35	23	11 ... 0	
050	Idle Accumulation for CPU0 and CPU1					
051 ... 056	Reserved					
057	CNTL PT For Move	Internal to MTR				CMCL
060	*1		CPU CNTL PT Assig	CPU0 Exchange Address		ACPL
061	*2		CP1 CNTL PT Assig	CPU1 Exchange Address		
062				Address of PP1 Exchange Package		PXPP
063	0 (P) =PPR		(A0) = 0		(B0) = 0	*5
064	CRM (LA), ON 6170	1073	LJM (LA) 0100	1073	CON 7773B 7773	SFPL "SFP" Load Code
065	PSN 0000	LCD RPLA (RPL Addr) 201	RPLA	CRM (LA), CM+3 6613	1073	Used to Find scope type PPU routines (see PPR section) Auto Load
066 ... 077	Reserved					
100	CH0 *3	CH1	CH2	CH3	CH4	CTIL
101	CH5	CH6	CH7	CH10	CH11	
102	CH12	CH13	CH14 FECT Unused	CH15 FNCT	CH16 FBCT	*7
103	CH17 Unused	CH20	CH21	CH22	CH23	
104	CH24	CH25	CH26	CH27	CH30	
105	CH31	CH32	CH33	CH34 Unused	CH35 Unused	
106	Seconds		Milliseconds			RTCL
107	Scratch	Input	Output	Rollout	LGO	MSAL RMS Storage Assign
110	Permanent File Activity Control *4					PFNL
111			Next time to activate *6			Removable device system control
112 ... 177	Reserved					*8 Used to designate designate specific devices for these files. If overflow, use any TEMP types device. If active bit 11 on & est ord in lower 6 bits.

*1 CPU 0 Off Flag (Bit 59)
*2 CPU 1 Off Flag (Bit 59)
*3 Channel Status

Bit 11, on Indicates Channel Requested
Bit 7-10, Unused
Bit 6, On,Channel Not Available
Bit 0-5, PP Assigned

* Descriptions follow:

Figure 2-2. Pointers, Constants, and Control Words (Continued)

*4	Bit 59	Total PF System Interlock
	Bit 58	Request Total PF System Interlock
	Bit 53-48	PF Activity Count
	Bit 47-18	Reserved
	Bit 17-12	Default Family Equipment Number
	Bit 11- 6	Alternate Family Count
	Bit 5-1	Reserved
	Bit 0	Word Interlock

*5 This is the first word of any Pool -PP- exchange package. A pool PP will read up this word at preset time during deadstart and store it into its own core for later use during an exchange jump.

*6 Delay for 1SP to call CMS. The PF activity portion of this word is updated by every copy of PFM and all PF utilities. In order to avoid conflict with this field, the word is interlocked via the SFBM monitor function, which sets bit 0. Each PP routine will clear this bit when it has incremented the PF activity count.

*7 See table 2-2 to call CMS.

*8 This word is used to designate specific devices for these types of files. If the device should fill up, then overflow to any TEMP device. If active, then bit 11 is set and bits 0-5 contains est ord, bits 6-10 are zero.

Figure 2-2. Pointers, Constants, and Control Words (Continued)

TABLE 2-7. CMR

This description corresponds to Figure 23-2 p. 23-18

Address	Byte	Description
0000		Always zero
0001	0, 1	FWA RPL = 20547
	2	Number of PPU's = 24
	3	0014 = 0000 0000 001 100 = bits 2^{14} and 2^{15} set. CMU and CEJ/MEJ present
	4	Memory size = 30000 = 98K system
0002	0, 1	FWA PLD = 34131
	3	Number of CP's = 27
	4	FWA PPO OR = 6200
0003	0	Dayfile pointers FWA = 6400
	1, 2	FWA dayfile dump buffer = 15470
	4	Number of excess dayfiles = 3 SYSTEM, ACCOUNT, and ERROR
0004	0	FNT FWA = 6700
	1	FNT LWA+1 = 7700
	3, 4	JBC FWA = 11420 FNT is 1000B words long or 400B files long
0005	0	EST FWA = 6600
	1	EST LWA+1 = 6700
	2	EST RMS LWA+1 = 6602 EST is 100 words long and at most there are 2 RMS devices
0006	0, 1	RCL FWA = 34130
0007	0, 1	CLD FWA = 34313
0010	0, 1	CLD for COS FWA = 35315
0020	4	CMR size is 35400
0032+35		System date line
0045	3	BATCHIO is at CP26
0045	4	MAGNET is at CP25 No other subsystems are active
0057	0	No CP is currently scheduled for a move

TABLE 2-7. CMR (Continued)

This description corresponds to Figure 23-2 p. 23-18

Address	Byte	Description
0060	2	CPU0 assigned to CP3 at CPA 600
	3, 4	CPU0 EPA at 600
0061	0	CPU1 is not available
0062	4	PP1's EPA is at 20043
0063	0, 1	PPR address in CPUMTR at 16736
0064 and 0065		Auto load code for SFP, see Chapter 4 on PP Resident
0076	3	Channel 10 is assigned to PP1, i.e., DSD has the display channel.
0102	0	Channel 12 is assigned to PP6
0221		Note that CP1 is an available CP since the JNMW word is all zero.

Word	Contents (bits 59 … 0; field boundaries at 47, 35, 23, 11)	Symbol
000 – 017	Exchange Package Area	
020	*1 | No PPU | Error Flags | Pseudo Activity *10 | REF Addr /100B | FL/100 B	STSW
021	Job Name | Job Orig | Operator Assig Equip *8	JNMW OAEW
022	CPU Priority | Queue Priority | *2 | *3 | *4 | Time Limit | CPU Allowable	JCIW
023	CM Resident Time Limit | *5 | CPU Time Slice Limit	TSCW
024	Seconds | Milliseconds	CTLW CPU Time Limit
025	PP Input Register / PP Recall Register	RLPW
026	Bit 12 is PP Pause Flag; Bits 6-11 are Sense Switches | *9	SNSW
027	Reserved	
030 – 034	Message 1 Area, 1st Line B Display	MS1W
035 – 037	Message 2 Area, 2nd Line B Display	MS2W
040 – 047	Installation Area	
050	Time *6 Limit Exceeded | | CPU Time (MS) Milliseconds	ACTW, CPTW
051	Start Time (CPU Seconds) | FL/100 * Time	CMUW
052	| Number of Sectors Transferred	MSUW
053	| Number of Physical Records Transferred	MTUW
054	CPU Time For Job | CPU Time (MS) Milliseconds	ACTWE, CPJW

* **Descriptions follow:**

Figure 2-3. Control Point Area * 7

* 1 W Bit 59 status is waiting for CPU; if Bit 58 X status is set the control point is in recall; if Bit 57 auto recall status is set CPU is in AUTO-recall. If Bit 56 is set, this control point has sub-control points active. Bit 53 is job advancement flag.

* 2 Bits 33, 34, 35 are CPU status for rollout. This is a copy of W, X, R from word 20 (STSW), so 1RI can set them properly on a subsequent rollin.

* 3 If Bit 27 is set, rollout is in process.

* 4 If Bit 24 is set, rollout is requested.

* 5 If Bit 35 is set, CPU time slice is active. If not set, time slice exceeded.

* 6 2000B is set.

* 7 To convert from control point number to control point area address, left shift control point number by 7, as shown for CP3 and 2400B following:

```
                 1st  2nd
     CP3         0110000000
                              3rd
                   6   0  OB

               Reverse    1st
     2400B 3rd  01010  0000000  B
                            2nd
                 1  2 CP
               4th
```

* 8 OAEW is the EST number assigned by the operator. PP routines (LFM) retrieves this information from this byte and clears it. In addition, whenever TCS is called to process a new control card this byte is cleared. This implies that equipments cannot be preassigned by the operator.

* 9 To make the message at MSIW on the B display flash, set the PP pause bit.

* 10 For every request to CIO for tape activity/per tape, this counter is bumped by 1. When 1MT completes activity, it decrements the counter by 1. Maximum activity is maximum number of drives (also see UADM). This is a tape activity count, not a tape number count.

Figure 2-3. Control Point Area * 7 (Continued)

Word	59	47	35	23	11 (to 0)	
055 ... 057	Reserved					
060	Job Card FL	Last Card FL	FL of Program Calling DMP=	Rollin FL	Increase FL Req	FLCW
061	*1	Alternate Library File Name			Map Cont.	LDCW
062	Equip Number	Reserved	Terminal Interrupt Address *10		Output Pointer	TIOW, TIAW
063	Auxiliary Pack Name (Default)			EST ORD of Family Device	*2	PFCW
064	User Number			*8	User Index	UIDW
065	400B =NO Exit Flag / Error Flag	Terminal Input Buffer			Error Exit Return Addr	EECW, TINW *3
066	Input Buffer Address		Right Screen Buff Address	Left Screen Display Address		DBAW, K Display Control
067	FST Addr of Input	*4 / Control Statement Count		Next Statement Index	Limit Index	CSPW
070	*5 / Equip Numb	First Track	Current Track	Current Sector	Sector Flag *9	
071	Job Sequence Number			Demand File Random Index		RFCW
072	Project Number 0-10 Characters With Zero Fill (Left Justified)					APJW
073	Max Mag Tapes	Max Disk Packs	Max MS Tracks	Max Work Files	Max Jobs (Batch)	APUW
074	Open Reserved		Max CPU Priority	Max Time Limit	Max FL	ACUW
075	Each Bit has Special Meaning					AACW
076	Length of BUF 0	Address of BUF 0		Length of BUF 1	Address of BUFF 1	ICAW Inter-Control Point Comm Area
077				Event Descriptor	Rollout Time Period	UPCW user profile
100	*6	Reserved	DMP= Parameter	SSJ= Parameter		SEPW
101	CP Prog Entry pt Name		Status Return	Parameter Block Addr		SPCW Bit 47 is special processor request active
102	EF	R3	R2		R1	JCRW *7
103 ... 127	Reserved					
130 ... 177	Control Statement Buffer					CSBW - 1/2 PRU (40B CM words) of Control Statement file.

Used for TELEX org jobs

* Descriptions follow:

Figure 2-3. Control Point Area (Continued)

* 1 If set, Bit 56 is no field length reduction flag.

* 2 The next three values indicate an index into a Table of Limits defined in COMSPFM.

Bits 6-8	Indirect access file size
Bits 3-5	Number of Permanent files
Bits 0-2	Max cumulative size of indirect files

* 3 Bit 47, set if error flag instead of error option.
Bit 18, field 0-17 is reprieve error return address.

* 4 If set, Bit 47 is EOR on control statement file.

* 5 If set, Bit 59 is information is for input file.
If set, Bit 58 is skip to exit flag.
Bit 59 indicates that CC's are being accessed either from file INPUT or some other file. For example, with procedure files, CONTROL will have created a new CC file, set this word to point to its current track/sector, and eliminate the FNT/FST for the new CC file. However, the tracks are left reserved so that this job can keep this new CC file but FNT space is cleared.

* 6 Special Entry Point Word

Bit 59, indicate presence of entry points
Bits 58-54, reserved
Bit 53 ARG = entry point present
Bit 52 DMP = entry point present
Bit 51 SDM = entry point present
Bit 50 SSJ = entry point present
Bit 49 VAL = entry point present

* 7 KRONOS CONTROL LANGUAGE registers and error flag (EF).

* 8 VAL= flag. = 1 VAL= SEP must be present in next program loaded via CC.
0 VAL= SEP not needed.

* 9 1st/2nd sector flag indicates to 1AJ which half of the sector of CC's is in the CSBW buffer.

* 10 See TSEM request VSDT and VCDT figure 13-12.

Figure 2-3. Control Point Area (Continued)

TABLE 2-8. EXAMPLE OF CPA AND RA + 0 THROUGH RA + 100

Foil No.	Address	Description
1	A	Dayfile of Job. Job ran at CP3 which is CPA=600B. The dump is taken at the CC.4, ABSDMP (600B, 1000B). Note that the job has requested and received one nine track tape.
	B	Job limits for CPA comparison
2	600 and 617	Exchange package
	600	P = 1732
	601	RA = 30700
	602	FL = 60000
	603	EM= 0007 all errors
	606	MA= 600
	620	Status field
	Byte 0	status = R, PPU No. = 1
	Byte 3	RA = 307 * 100
	4	FL = 600 * 100
	621	Name = MORRABWA Q Priority = 4010
	623	CM Time limit = 3222 CPU Time slice limit = 40000
	624	Time limit = 1 sec
2	630 and 637	Message 1 and 2 area = 4, ABSDMP (600B, 1000B) which is image of CC.
	650 and 654	Accounting information
	650	CPU time = 63
	651	Start time = 61, FL/100* time = 2
	652	Number of sectors transferred = 23
	653 and 654	Job is not active so these fields = 0
	660	Job card FL = 60000. Last card FL was job card = 60000. FL for DMP=call = 10000 from RESEX for REQUEST card. DMP= EQU 10000B, rollin FL = 0 job has not been rolled in, FL FL increase request = 0.
	661	from LIBRARY card = MORRIE

TABLE 2-8. EXAMPLE OF CPA AND RA + 0 THROUGH RA + 100 (Continued)

Foil No.	Address	Description
	663	IPF cumulative size = 7 FC, PF limit = 5 = CS. IPF single size = 7 = FS. From old lev C of Install Handbook: 7 = unlimited, 5 = 50000B = 204800D compare to limits on foil 1
	664	UN = MLO, UI = 1
	667 and 670	INPUT file CC control
	667	FST address = 3313, CC Count = 12 Next statement index = 157, limit index = 170
	670	Bit 59 set = information is INPUT file, eq no = 0, 1st track = cur track = 4302.
2		Note: Sector no. = 1, flag = 1 indicates 1st half of sector, eor flag in word 667 is not set since CC's continue in the 2nd half of the sector.
	671	Job seq No. = AABW Demand file rand index = 1 for assigned tape
	673 and 675	Validation info from VALIDEX MT = 4, RP = 4, MS = 1750B = 1000D. Loc files = 628 = 50D, DB = 12B = 10D. PR = 77, TL = 7777, FL = 1071, AW = 7——7. compare to limits in foil 1
	702	R2 = 1 from CC SET R2 = 1
	730	One half pru of CC
	600 + 157 + 757	Next statement index ⟶ DMP (0, 1000)
	Note 754	is last statement = current statement
	600 + 170 = 770	Limit of CC or LWA + 1 of CC
3	1	END, job is done
	2	1st arg 600B
	3	2nd arg 1000B
	4	Zero word to indicate end of arguments
	64	CC which called this routine = ABSDMP with 2 args
	65	CMU present next word for load 1761 note that P = 1732 from CPA so SYS= "XJ" must be at 1731.
3	66	CEJ/MEJ present job origin = 01 = BATCH. 1st word of object program is at 100.

TABLE 2-8. EXAMPLE OF CPA AND RA + 0 THROUGH RA + 100 (Continued)

Foil No.	Address	Description
	70 and 77	CC image ABSDMP (600B, 1000B)
	100	Entry point address of this overlay = 1567 and LWA + 1 of last and largest overlay, in this case this is the only (0, 0) overlay, = 1761. Note word 65, next word avial for load is also 1761.

```
MORRAAW.   74/08/27. MORRIES PERSONAL KRONOS 2.1.

09.36.26.MORRI.CM60000.
09.36.26.ACCOUNT.MLO..
09.36.27.RESOURC(NT=1)
09.36.27.COMMENT.  THIS IS A TEST OF CONTROL CARD
09.36.27.S.
09.36.27.REQUEST.A.
09.36.56.NT61. ASSIGNED TO A       . VSN=****61.
09.36.56.CATLIST.
09.36.56. CATLIST COMPLETE.
09.36.57.GET.MLOPL.
09.36.58.SET(R2 = 1)
09.36.58.LIBRARY(MORRIE)
09.36.58.4.ABSDMP(6000.1000B)
09.36.59.ABSOLUTE DUMP COMPLETE
09.36.59.DMP(0.1000)
09.37.00.LIMITS.
09.37.00.SETTL.777.
09.37.00.COMMENT.  THIS IS ABOUT THE END OF THIS
09.37.00.TEST
09.37.00.RETURN.A.
09.37.00.DISPLAY EF.
09.37.00. FORMAT ERROR ON CONTROL CARD.
09.37.00.EXIT.
09.37.00.COMMENT.  THIS IS AN ERROR
09.37.01.COMPASS.
09.37.01. IDENT CARD MISSING.
09.37.01.          1 WARNING MESSAGE IN  CARD
09.37.01.          1 ERROR IN  CARD
09.37.01. ASSEMBLY ERRORS.      44500B  CM USED.
09.37.01.     0.241  CPU SECONDS  ASSEMBLY TIME.
09.37.01.CP        0.394 SEC.
09.37.01.CM        0.002 KWH.
09.37.01.MS        0.086 KPH.
09.37.26.LP20      0.320 KLN.
```

(A)

Snap shot

Listing of Control cards

```
MORRI.CM60000.
ACCOUNT.MLO.MLO.
RESOURC(NT=1)
COMMENT.  THIS IS A TEST OF CONTROL CARDS.
REQUEST.A.
CATLIST.
GET.MLOPL.
SET(R2 = 1)
LIBRARY(MORRIE)
4.ABSDMP(6000B.1000B)
DMP(0.1000)
LIMITS.
SETTL.777.
COMMENT.  THIS IS ABOUT THE END OF THIS TEST
RETURN.A.
DISPLAY EF.
EXIT.
COMMENT.  THIS IS AN ERROR
COMPASS.
```

```
          LIMITS.                          74/08/27. 09.37.00.      PAGE     1

     MLO              1   74/06/17. 74/07/18.

         AH =.
         AB =.
         AB =.
         AB =.
         RT =        4.
         RP =        4.
         TL =    7777B.
         PR =      77M.
         CM =   32000B.
         NF =       50.
         MS =     1000.
         PN =.
         DB =       10.
         FC =  UNLIMITED.
         CS =    20480.
         FS =  UNLIMITED.
         MA = EVEN     .
         RO =    SYSTEM.
         PX = HALF     .
         TT = TTY      .
         AW = 77777777777777777777
```

(B)

ABSOLUTE DUMP FROM 000600 TO 001000 PAGE 1

2

000600	00001732555555000000	0Z	00030700000001000001	CG A A	00040000000155777717	F A I:0
000603	00070000000153000000	G AS	00000000000152000002	A) B	00000000000154000002	A= B
000606	00000600000001000012	F A J	00000000000161000001	A(A	00000000000000000055	
000611	00000000000000000000		00000000000000001000	H	17274000000000000000	O=5
000614	00000000000000000001	A	17275260000000000000	O=)≡	24142020000000000160	TLPP A≡
000617	33333433333300000000	G01000	10010000000003070600	HA CGF	15172222010227010000	MORRABWA
000622	00031401000000000000	X5H	00003222000000040000	ZR D	00000100000000175000	A O/
000625	00000000000000000000		00000000000000000000		00000000000000000000	
000630	37560102230415205141	4,ABSDMP(6	33330256343333330252	00B,1000P)	00000000000000000000	
000633	00000000000000000000		00000000000000000000		00000000000000000000	
000636	00000000000000000000		00000000000000000000		00000000000000000000	
000641	00000000000000000000		00000000000000000000		00000000000000000000	
000644	00000000000000000000		00000000000000000000		00000000000000000000	
000647	00000000000000000000		00000000000000000063	I	00000000610000000002	[B
000652	00000000000000000023	S	00000000000000000000		00000000000000000000	
000655	00000000000000000000		00000000000000000000		00000000000000000000	
000660	06000600010000000000	F F A	00151722221103000000	MORRIE	00000000000000000000	
000663	00000000000000000757	G.	15141700000000000001	MLO A	00000000000000000000	
000666	00000000000000000000		33130000001201570170	OK JA.A+	40004302430200010001	5 8B88 A A
000671	01010227000000000001	AAB= A	00000000000000000000		00040004175000620012	D DO/] J
000674	07573700077777771071	G.4 IIIM+	77777777777777777777	IIIIIIIIII	00000000000000000000	
000677	00000000000000000000		00000000000000000000		00000000000000000000	
000702	00000000000001000000	A	00000000000000000000		00000000000000000000	
000705	00000000000000000000		00000000000000000000		00000000000000000000	
000710	00000000000000000000		00000000000000000000		00000000000000000000	
000713	00000000000000000000		00000000000000000000		00000000000000000000	
000716	00000000000000000000		00000000000000000000		00000000000000000000	
000721	00000000000000000000		00000000000000000000		00000000000000000000	
000724	00000000000000000000		00000000000000000000		00000000000000000000	
000727	00000000000000000000		15172222115603154133	MORRI,CM60	33333357000000000000	000.
000732	01030317251624561514	ACCOUNT,ML	17561514175700000000	O,MLO,	22052317252203511624	RESOURC:NT
000735	54345255000000000000	=1)	03171515051624575555	COMMENT.	24101123551123550155	THIS IS A
000740	24052324551706550317	TEST OF CO	16242217145503012204	NTROL CARD	23570000000000000000	S.
000743	22052125052324560157	REQUEST,A.	00000000000000000000		03012414112324570000	CATLIST,
000746	07052456151417201457	GET,MLOPL,	00000000000000000000		23052451223555545534	SET(R2 = 1
000751 C	52550000000000000000	)	14110222012231511517	LIBRARY(MO	22271105525500000000	RRIE)
000754 N	37560102230415205141	4,ABSUMP(6	33330256343333330252	00B,1000A)	00000000000000000000	
000757	04152051335634333333	DMP(0,1000	52550000000000000000	)	14111511242357550000	LIMITS.
000762	23052424145642424257	SETTL,777.	00000000000000000000		03171515051624575555	COMMENT.
000765	24101123551123550102	THIS IS AB	17252455241005550516	OUT THE EN	04551706552410112355	D OF THIS
000770	00000000000000000000		00000000000000000000		00000000000000000000	
000773	00000000000000000000		00000000000000000000		00000000000000000000	
000776	00000000000000000000		00000000000000000000		00000262000400000000	BJ O
001001	00110700000001000001	IG A A	00000000000064000002	≠ B	00070000000000000000	G
001004	00000000000000000000		00000000000002000000	B	00001000000001000000	H A
001007	00000000000000000000		77777777777777777700	IIIIIIIII	00000000000000000000	
001012	00000000000230000000	S	00000000000000000007	G	00000000000000000000	
001015	00000000000000000000		05160420000000000000	ENDP	00000000000000000012	J
001020	00000000000011070000	IG	00000000000000000000		00000000000000000000	
001023	00000473000000010000	D> A	77777777770000000000	IIIII	00000000000000000000	
001026	00000000000000000000		00000000000000000000		00000000000000000000	
001031	57333335554230503570	.002 SEC.	00000000000000000000		00000000000000000000	
001034	00000000000000000000		00000000000000000000		23201401315755555104	SPLAY. (D
001037	11235200000001005614	15) A .L	00000000000000000000		00000000000000000000	
001042	00000000000000000000		00000000000000000000		00000000000000000000	
001045	00000000000000000000		00000000000000000000		00000000000000000000	
001050	55555555555533573335	0.002	00000000000000000000		00000000000000000000	
001053	00000000000000000000		00000000000000000000		00000000000000000000	

2-16.6

E N D

DUMP FROM	0 TO 1000															
0	00000	00000	00000	00000	05150	42000	00000	00000 END	41333	30200	00000	00000 600B	34333	33302	00000	00000 1000B
4	00000	00000	00000	00000	34213	42214	01342	33051	12372	30010	00344	43043	13771	10134	43305	71701
10	62403	14313	77344	30100	11130	20016	65200	21537	03140	20016	65020	01356	04051	40034	57010	01113
14	20031	53754	00111	21003	54001	10301	00110	20200	16652	00116	31036	40200	16650	20013	56055	12003
20	00410	33106	24000	02313	77770	15300	01530	00153	00000	00000	00000	00000	00000	00000	00000	00000
24	42604	00042	61002	04202	70420	00000	00000	30005	15172	22211	00004	00377	00000	00000	00000	00000
30	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
	DUPLICATED LINES.															
44	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	01300
50	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
	DUPLICATED LINES.															
60	42374	10240	03400	44017	40054	00640	07401	00017	40114	01240	13401	40017	40154	01640	17402	00017
64	01022	30415	20000	00002 ABS DMP	40000	00000	00000	01761	40121	40000	01000	00100	00000	00000	00000	00000
70	01022	30415	20514	13333	02563	43333	33025	20000	00000	00000	00000	00000	00000	00000	00000	00000
74	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
100	00000	00000	17610	01567	22227	36727	20603	15213	20701	12661	36771	63525	20506	15150	63410	22244
104	73113	11505	03320	00171	66357	04400	00110	66322	03150	00111	04430	00110	67545	05520	00111	64300
110	43400	14300	00111	10677	04000	01610	00000	00000	71400	07774	43066	76600	66211	13777	61307	77744
114	20425	66500	71300	00017	04000	00103	61000	46000	00000	00000	61000	46000	51400	00124	61300	00006
120	43271	66200	61407	77755	15712	20466	66223	73374	21103	36443	03110	00121	20466	22624	04000	00116
124	55555	55555	55555	55555	42612	40112	26701	20466	66224	40463	73663	36464	03170	00125	20466	22624
130	00000	00000	61000	46000	51200	00134	54321	27101	66200	54431	61307	77755	61400	00006	04000	00125
134	17170	63146	31463	14632	20030	00000	00000	00012	55555	55555	55555	55555	04000	01710	00000	00000
140	51200	00151	11721	201 3	11621	20103	20733	11021	20625	20103	36767	11521	20103	20017	11621	36770
144	20511	20603	36776	54421	54341	12775	11274	54531	11173	20736	36252	11074	36005	20155	11373	36712
150	20355	36603	04000	00137	70000	70000	70000	70000	07070	70707	00000	00000	00000	07070	70707	00000
154	33333	33433	33333	33333	00000	00000	00000	01000	55051	52624	31555	55555	00000	00000	00000	00001
160	77752	42025	24000	00001	17252	42025	24000	00035	00000	00000	00000	00166	00000	00000	00000	01566
164	00000	00000	00000	01566	00000	00000	01000	01567	34010	22417	14252	40555	04251	52055	06221	71555
170	33333	34133	33555	55555	55241	75555	55555	55555	33333	43333	33555	55555	55555	52001	07055	55555
174	55555	55555	34000	00000	00000	00000	00000	00000	55553	33333	41333	35555	55553	33333	33344	23635
200	40404	04040	40333	33333	33335	55555	55555	51732	55555	50055	55555	55555	33333	33633	42333	33333
204	33333	33433	33333	33334	55555	55500	03070	05555	01000	00155	55555	53333	33413	33333	33333	33334
210	40404	24242	42344	25555	55550	00655	55000	15577	77170	00000	00000	00000	55553	33333	41333	65555
214	55543	33333	42333	33333	33333	33440	36333	33333	33335	55555	55000	75555	00015	30055	55555	55555
220	33333	33333	33333	33333	33344	03533	33333	33335	55555	55555	55555	50001	42000	00255	55555	53333
224	33333	33333	33333	33334	40373	33333	33333	55555	55555	55555	55000	15400	00020	00000	00000	00000
230	55553	33333	41334	15555	55553	33333	33334	13333	33333	33333	34333	33333	34355	55555	55555	50600
234	55550	10000	12555	55555	33333	33333	33333	33333	33344	13433	33333	33334	55555	55555	55555	50001
240	61000	00155	55555	33333	33333	33333	33333	33333	33333	33333	33404	05555	55555	55555	55555	55555
244	00550	00000	00000	00000	55553	33333	41343	45555	55553	33333	33333	33333	33333	33333	33333	33333
250	33335	55555	55555	55555	55555	55555	55555	55555	33333	33333	33333	33333	33333	33333	33343	33333
254	55555	55555	55555	55555	55551	00055	55555	53442	35423	73333	33333	33333	33333	33333	33333	35555
260	55551	72740	00555	55555	55550	00000	00000	00000	55553	33333	41343	75555	55553	33333	33333	33333
264	33333	33333	33333	33333	33345	55555	55555	55555	55555	55500	01555	55555	34423	54240	35413	33333
270	33333	33333	33333	33333	55555	55517	27526	05555	55555	55555	55555	53537	34373	53335	33333	33333
274	33333	33333	34413	35555	55552	41420	20555	55555	01600	00000	00000	00000	55553	33333	41344	25555
300	55553	03636	36363	73636	36363	63633	33333	33333	33335	55555	55333	33433	33335	55555	55555	55555
304	34333	33433	33333	33333	33333	33633	42334	13333	55555	55510	01555	55555	03070	60055	55555	53440
310	34423	53535	35333	43335	35423	33433	33333	35555	55551	51722	22010	22701	55550	00000	00000	00000
314	55553	33333	41353	55555	55553	33336	33373	33433	33333	33333	33333	33333	33335	55555	55003	04010
320	55555	55555	55555	55555	33333	33336	35353	53333	33333	33333	37333	33333	55555	55555	55322	25555
324	00045	55555	55555	53333	33333	33433	33333	13333	33333	44240	33333	35555	55555	55501	00555	50017
330	50000	00000	00000	00000	55553	33333	41354	05555	55553	33433	33333	33333	33333	33333	33333	33333
334	33335	55555	55555	55555	55555	55555	55555	55555	33333	33333	33333	33333	33333	33333	33333	33333
340	55555	55555	55555	55555	55555	55555	55555	53333	33333	33333	33333	33333	33333	33333	33333	35555
344	55555	55555	55555	55555	55550	00000	00000	00000	55553	33333	41363	35555	55553	64240	41333	43335
350	35363	33734	40353	34034	37345	55555	55375	60102	23041	52051	41555	55555	36363	63633	35404	13637
354	36363	63636	36333	54035	55555	55533	33025	63433	33330	25555	55555	53333	33333	33333	33333	33333
360	33333	33333	33333	35555	55555	55555	55555	55555	55550	00000	00000	00000	55553	33333	41363	65555
364	55553	33333	33333	33333	33333	33333	33333	33333	33335	55555	55555	55555	55555	55555	55555	55555

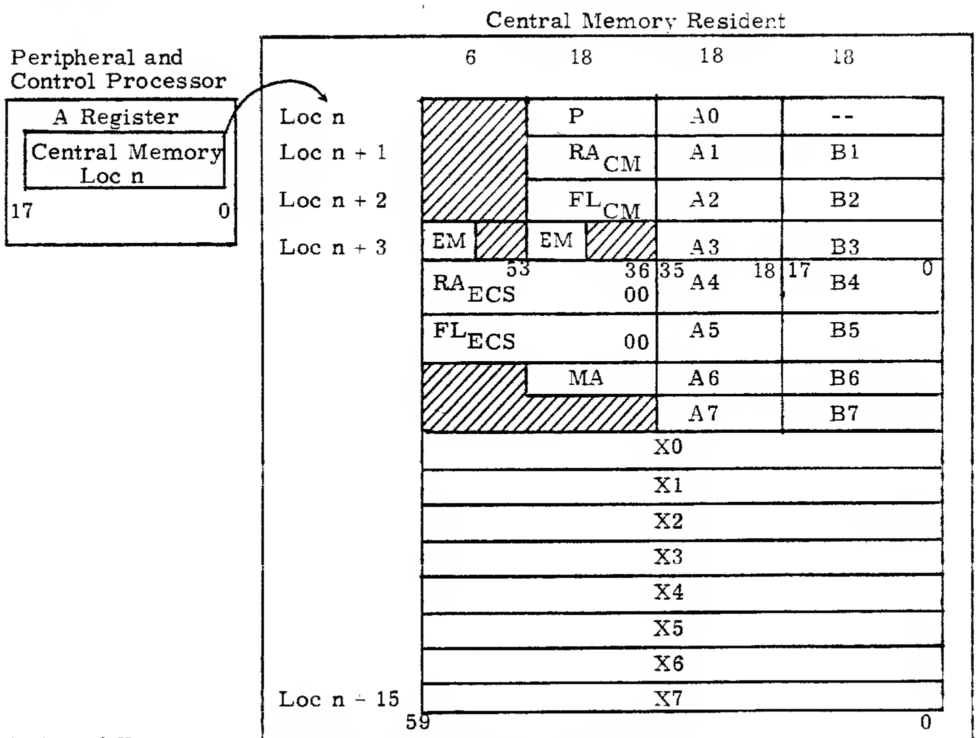
3

97404700C

2.1 EXCHANGE JUMP

An Exchange Jump instruction (XJ) starts or interrupts the central processor and provides central memory with the first address of a 16-word package in central memory. The Exchange Jump package (Figure 2-4) provides the following information on a program to be executed:

1. Program address (P)
2. Reference Address for Central Memory (RA_{CM})
3. Field length of program for Central Memory (FL_{CM})
4. Reference Address for Extended Core Storage (RA_{ECS})
5. Field length of program for Extended Core Storage (FL_{ECS})
6. Program exit mode (EM)
7. Initial contents of the eight A registers
8. Initial contents of the eight X registers
9. Initial contents of B registers B1 - B7 (B0 is fixed at 0)
10. Monitor Address (MA)



Descriptions follow:

Figure 2-4. Exchange Jump Package

MA	=	Monitor Address				
P	=	Program Address		A	=	Address Registers
RA	=	Reference Address		B	=	Increment Registers
FL	=	Field Length		X	=	Operand Registers
EM	=	Exit Mode =	000000	Disable Exit Mode		
		Octal Contents of Bits 36-53, Location "n +3"	010000	Address Out of Range		
			020000	Operand Out of Range		
			030000	Address or Operand Out of Range		
			040000	Indefinite Operand		
			050000	Indefinite Operand or Address Out of Range		
			060000	Indefinite Operand or Operand Out of Range		
			070000	Indefinite Operand or Address Out of Range or Operand Out of Range		

Figure 2-4. Exchange Jump Package (Continued)

The central processor enters the information about a new program into the appropriate registers and stores the corresponding and current information from the interrupted program at the same 16 locations in central memory. Hence, the controlling information for two programs is exchanged. A later Exchange Jump may return an interrupted program to the central processor for completion. The normal operation of the A and X registers is not active during the Exchange Jump so that the new entries in A are not reflected changes in X.

When an Exchange Jump interrupts the central processor, several steps occur to ensure leaving the interrupted program in a usable state for re-entry:

1. Instruction retrieval stops after all instructions from the current instruction word have been read.
2. The Program Address register, P, is set to the address of the next instruction word.
3. The instructions are performed.
4. The parameters for the two programs are exchanged.

A subsequent Exchange Jump can then re-enter the interrupted program at the point at which it was interrupted, with no loss of program continuity.

When CPU is in monitor mode:

XJ K + B_j B_j+K points to Exchange package area

When CPU is not in monitor mode:

XJ Then (MA) points to Exchange package area
Any B_i-designation is ignored.

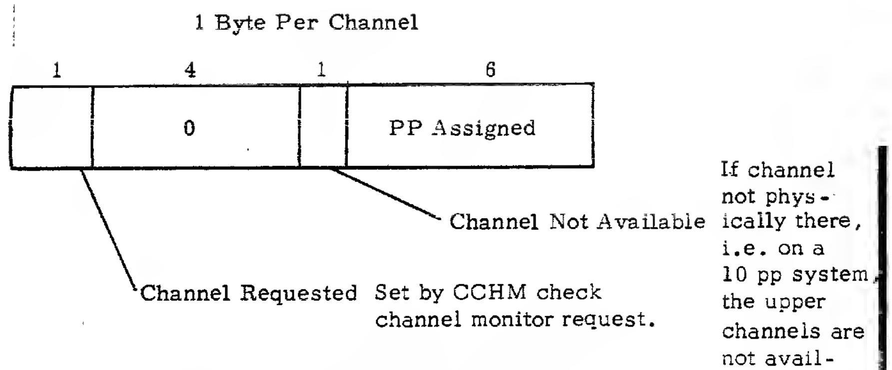


Figure 2-5. Channel Status Table (CST)

24		12	12	12
FWA Dayfile Buffer		(HOW FULL) Number of Words in Buffer	Length of Buffer (BUF SIZE)	0
Eq. No. For Dayfile	1st Track	Current Track	Current Sector	0
12	12	12	12	12

One entry for each system dayfile (Normal, Account, and Error)
One entry for each Control Point

Figure 2-6. Dayfile Buffer Pointers

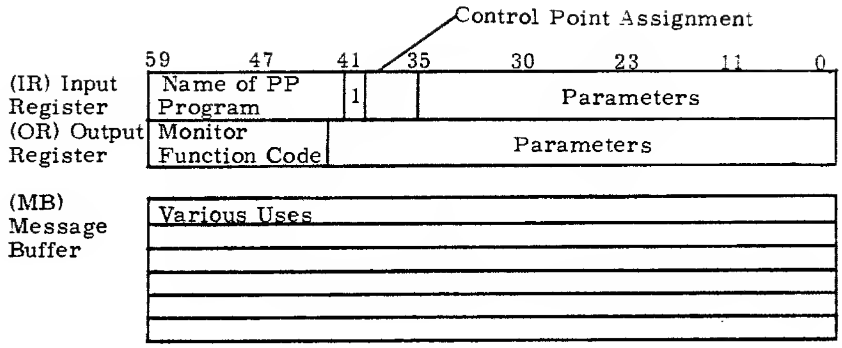
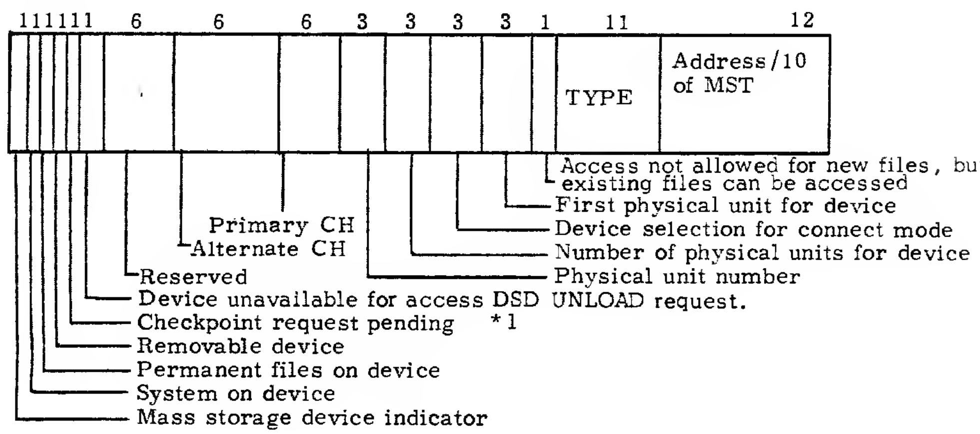


Figure 2-7. PP Communication Area

Mass Storage Devices:



Non-Mass Storage Devices:
(3000 Type Equipments)

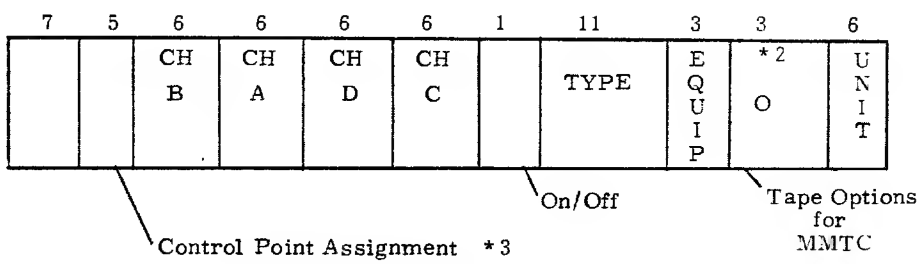


Figure 2-8. Equipment Status Table

*1 The checkpoint requested bit in the EST is set by:

The CPR (checkpoint requested bit) set by any PP directly, i.e., no monitor request, whenever it decides that the TRT should be saved on the device. There are two ways to set the bit. Either a PP uses the common deck COMPCKP or does it itself. The PP routines which use the preferred method COMPCKP are:

PFM. Any change in the TRT, write, modify, append, extend, purge, etc.

CIO. Only when a DA (direct access) PF has changed and forced a change in the TRT. Note: local files are not checkpointed specifically, only when some other action forces a checkpoint.

IMS. When initializing MS or EDIT DA files. Note: EDIT DA function 2 will purge any DAs whose UI is specified in a MASK. (No users currently).

The PP's who set the bit themselves are: (This may not be a complete list).

ORP. Whenever it releases a PF from a CP which has just written on it or it is purged and this is the last user of the file.

PFU. When requested (function 14) by PFLOAD in routine EOL at end of load if any files were copied onto the device.

*2 5 bits described in COMSMTX HP option under UDT description. See chapter 9.

*3 Set to 37B if job using this eq is rolled out.

Figure 2-8. Equipment Status Table (Continued)

An example of an EST is shown below.

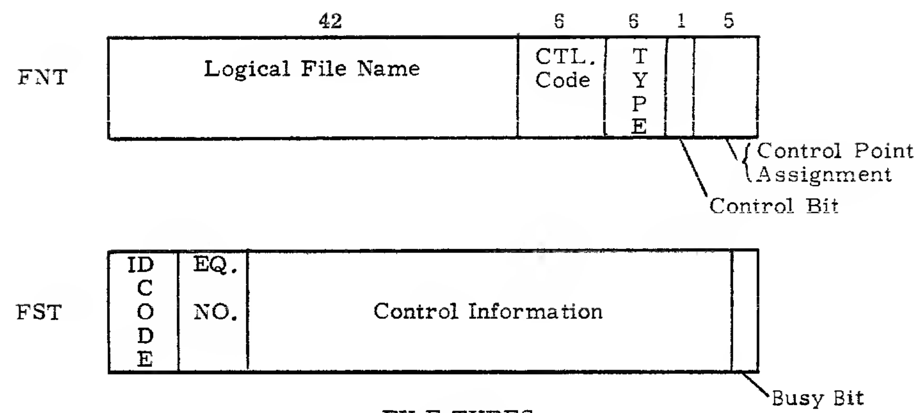
```
ABSOLUTE DUMP FROM 003200      TO       003300       PAGE        1               EST

003200  T0000002000104110430  + B AD10X  45000002000044110515  + B 9IEM  00000000000000000000
003203  00000000000000000000             00000000000000000000            00000000000000000000
003206  00000000000000000000             00000000000000000000            00000010000004237000  H DS+
003211  00000012000003224000    J CR5    00000000000000000000            00000000000000000000
003214  00000000000000000000             00000000000000000000            00000000000000000000
00321T  00000000000000000000             00000012000014205000  J LP/     00000000000000000000
003222  00000000000000000000             00000000000000000000            00000000000000000000
003225  00000000000000000000             00000000000000000000            00000000000000000000
003230  00000003000063247000    C IT+    00000000000000000000            00000000000000000000
003233  00000000000000000000             00000000000000000000            00000000000000000000
003236  00000000000000000000             00000000000000000000            20000003002064247000  P C P*T.
003241  20000111604064240000  P A1 5*T   00000000000000000000            00000000000000000000
003244  00000000000000000000             00000000000000000000            00000000000000000000
003247  00000000000000000000             00000013000015246000  K MT≡     00000013000015246001  K MT≡A
003252  00000013000055246000    K T≡     00000013000055246001  K T≡A     00000000000000000000
003255  00000000000000000000             00000000000000000000            00000000000000000000
003260  00000013000016245000    K NT/    00000013000016245001  K NT/A    00000000000000000000
003263  00000000000000000000             00000000000000000000            00000000000000000000
003266  00000000000000000000             00000000000000000000            00000606000623037001  FF FSC+A
003271  00000606000623037002  FF FSC+B   00000000000000000000            00000000000000000000
003274  00000000000000000000             00000000000000000000            00000000000002405000  IE
```

This is the CMR DECK which created the above EST.

```
CMRDECK
NAME= MORRIES PERSONAL KRONOS 2.1.
EQ0=DI-1.ON.0.1.2.
PF=0.D.377.MORRI.40.
RTK=0.C107.T12.S00.
TK=0.C107.T12.S01.
TK=0.C436.T13.S21.
TK=0.C437.T13.S21.
TK=0.C442.T13.S21.
EQ1=DI-1.OFF.0.0.2.
REMOVE=1.
EQ10=DS.ON.7.0.10.
EQ11=CR.ON.4.0.12.
EQ20=LP.ON.5.0.12.
EQ30=ST.OFF.T..3.
EQ40=TT.OFF.T..3..20.
EQ41=TT.OFF.0..11.1.40.
EQ50=MT-2.ON.6.0.13.
EQ52=MT-2.OFF.6.0.13.
EQ60=NT-2.ON.5.0.13.
EQ70=SC.ON.T.1.6.6.6.
EQ71=SC.ON.7.2.6.6.6.
```

Figure 2-8. Equipment Status Table (Continued)



FILE TYPES

SYSTEM SYMBOLS	CODE	TYPE
INFT	00	Input
ROFT	01	Rollout
PRFT	02	Print
PHFT	03	Punch
TEFT	04	Timed/Event Rollout Queue
SYFT	05	System
LOFT	06	Local
CMFT	07	Common
LIFT	10	Library
PTFT	11	Primary Terminal
PMFT	12	Direct Access Permanent File
FAFT	13	Fast Attach Files

Figure 2-9. General FNT/FST Entry Format

	6	6	1	5
JOBNAME	Job Origin	INFT	1	0

System Sector Contains Control Information

6	6	12	12	12	12
ID Code	EQ. No.	1st Track	0	FL/100B	Queue Priority

Figure 2-10. Input Queue Files

42	6	6	1	5
JOBNAME	Job Origin	PRFT	1	0

6	6	12	12	12	12
ID Code	EQ. No.	1st Track	1st Track Dayfile	1st Sector of Dayfile	Queue Priority

Can be changed by SETID card. (e.g. if a user wishes to route his output to a particular printer)

The DSD command is:
LPXX, YY
Where XX = EST and
YY = id #

Figure 2-11. Print Queue Files

42	6	6	1	5
JOB NAME	Job Orign	PHFT	1	0

6	6	12	12	12	12
ID. Code	EQ. No.	1st Track	0	Format See COMSJOT ON OPL	Queue Priority

Figure 2-12. Files in Punch Queue

42	6	6	1	5
JOBNAME	Job Origin	ROFT	1	0

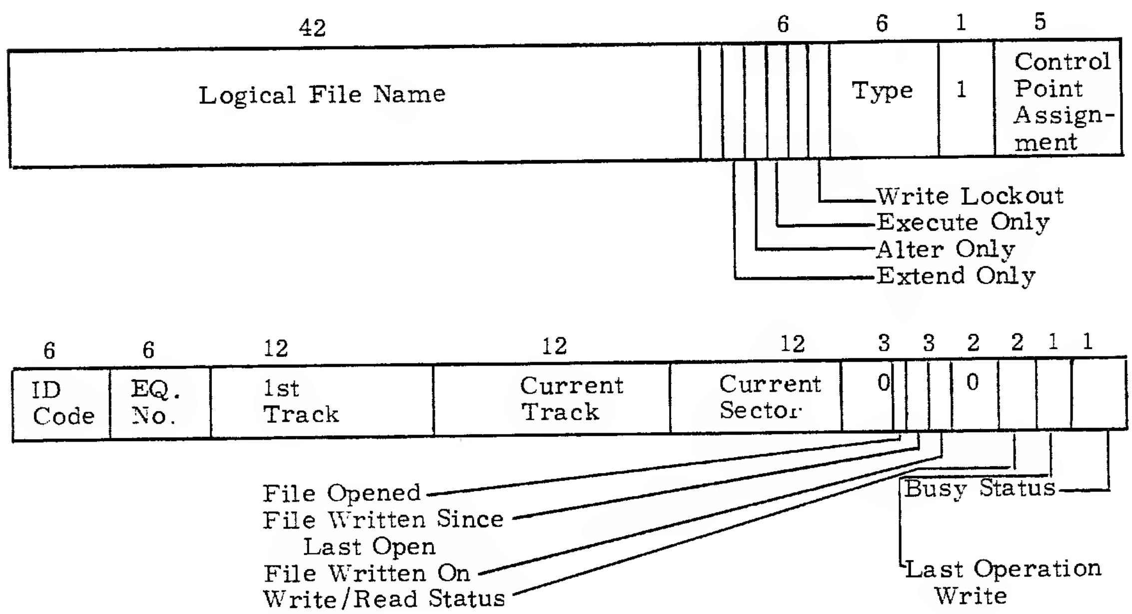
6	6	12	12	12	12
ID Code	EQ. No.	1st Track	0	FL/100B	Queue Priority

Figure 2-13. Files In Rollout Queue

42	6	6	1	5
JOBNAME	Job Origin	TEFT	1	0

6	6	12	12	12	12
Job Descriptor	EQ. No.	1st Track	Event Descriptor	FL	Rollout Time Period

Figure 2-14. Files in Timed/Event Rollout Queue



0 = incomplete Read/Write
1 = EOR
2 = EOF
3 = EOI

If last operation was write or position, bit 2 = operation complete = IN = OUT = FIRST

Figure 2-15. Mass Storage Files - Assigned to a Job or Common

In order to get multi-user read on locked COMMON files, every user who requests this file gets a separate FNT/FST pointing to the file and its type is set to LIBRARY.

Multiple fast attach files (FA) are handled by PFM or any PP, if more than one family is on the system, in the following manner:

FA files are accessed by PFM for CP routines and PFM will generate a local FNT/FST copy for the user. It is faster, since PFM always checks for FA status. PP routines can check the eq number of the file in the FNT/FST and determine if this device resides in the family which it is accessing.

42	4	1	1	6	1	5
Logical File Name	0			Type	0	CTL. PT.

Write Lockout
Opened

Address in MAGNET FL used by CIO
to send the 3 word control for a tape request
to MAGNET

6	6	12	4	2	18	1	11
ID. Code	EQ. No.	UDT Address of Assigned Tape			VSN Entry Random Address		Same as for Mass Storage Files Status Field

Type $\begin{cases} 0 = \text{VSN Entry} \\ 1 = \text{Seven Track} \\ 2 = \text{Nine Track} \end{cases}$
Format

Figure 2-16. Magnetic Tape Files

42	1	1	1	1	1	1	6	6
Logical File Name							FAFT	Control Point Assignment

Read
Write
Execute
Append
Modify

6	6	12	12	6	6	12
ID Code	EQ No.	1st Track	User Count for Read, Allow, Modify Access	User Count for Read Allow Append Access	User Count for Read Access	Same as for Mass Storage Files Status Field

Figure 2-17. Fast Attach Permanent Files

Fast attach files are permanent files that have been given FA status by the ISF Initialize System Files. ISF generates FA status for RESEXDF, RESEXVF, and VALIDUX, PROFILO for each family on the system. The FA technique is designed to achieve fast access to certain PFS, since PFM checks for FA status on a PF first.

Figure 2-17. Fast Attach Permanent Files (Continued)

The following is an example of an FNT.

FNT/FST

```
ABSOLUTE DUMP FROM 003300        TO        003340        PAGE        1

003300   23312324051500010700   SYSTEM AG   00000001400100010005   SASA A E     26011411042530001300   VALIDUX K
003303   00000264000000000005    7*      E   23011426012205000701   SALVARE GA   00000576457600020315      ---- RCM
003306   22052305300460001300   RESEXDF K   00000266000000000005    7V      E   22052305302406001300   RESEXVF K
003311   00000267000000000005    7A      E   11162025244700010011   INPUT* A I   00000612461200020005      -J-J R E
003314   04251520010573000200   DUMPAES B   00000615462100210410    -M-Q QNH    34152410010420000610   IMTHADP FH
003317   00000000000000000001           A   17252420252400000211   OUTPUT  A.   00000622462200020707      -R-R RGG
003322   00000000000000000000               00000000000000000000                00000000000000000000
```

ACTIVE FILES

Job Name	FNT Number	TYPE	FILE Name	1st Track	C.P.	
None	0	CMFT = 7	SYSTEM	1	0	Note that the write lockout bit is set. Hence, this is a locked common file.
None	1	FAFT = 13	VALIDUX	264	0	
TELEX	2	CMFT = 7	SALVARE	576	1	Note that the write lockout bit is not set. Hence, this is an unlocked common file.
None	3	FAFT = 13	RESEXDF	266	0	
None	4	FAFT = 13	RESEXVF	267	0	
DIS	5	INFT = 0	INPUT*	612	11	
Output Queue	6	PRFT = 2	DUMPAES	615	0	
MAGNET	7	LOFT = 6	IMTHADP	None	10	
DIS	10	PRFT = 2	OUTPUT	622	11	

CMFT = common
FAFT = fast attach
PRFT = print
LOFT = local

TRTL TRT Definition

	24	12	12	12
MST + 0	Number of Available PRU's (Minimum) * 1	See Table 7-1 Length of TRT in CM Words	Reserved	Number of Available Tracks left to assign

MSDL Mass Storage Driver Words

	12	1	1	1	9	18	18
MST + 1	Current Position of device (808, 6603 only)				Single Unit Sector Limit	Maximum Sector Limit/Trk	Minimum Sector Limit/Per Trk

- Reserved
- Release Reservation When Channel Released
- Format Pack Request Pending (844 Only)

MST + 2	Reserved For Mass Storage Drivers	* 2
MST + 3	Reserved For Mass Storage Drivers	

DEVL Device Allocation Information

	12	12	12	12	12
MST + 4	First Track of Indirect Access Files	Label Track (Linked to * 3 Catalog Tracks for P.F. Master Device)	1st Track of Permission Information	Actual Number of Catalog Tracks	System Table Track * 4

Figure 2-18. (MST) Mass Storage Table

*1 Minimum PRUs available is No of tracks avail * min sector limit. This value is computed by CPUMTR so that PP's don't have to perform the calculation. It is used when deciding if a particular file will exceed the device size.

*2 Used for dumping messages for error recovery.

*3 See PFDL MST + 6 word.

*4 A copy of CMR is written on this track when CHECKPOINT SYSTEM is requested. It is used for Level 1 and 2 recovery.

Figure 2-18. (MST) Mass Storage Table (Continued)

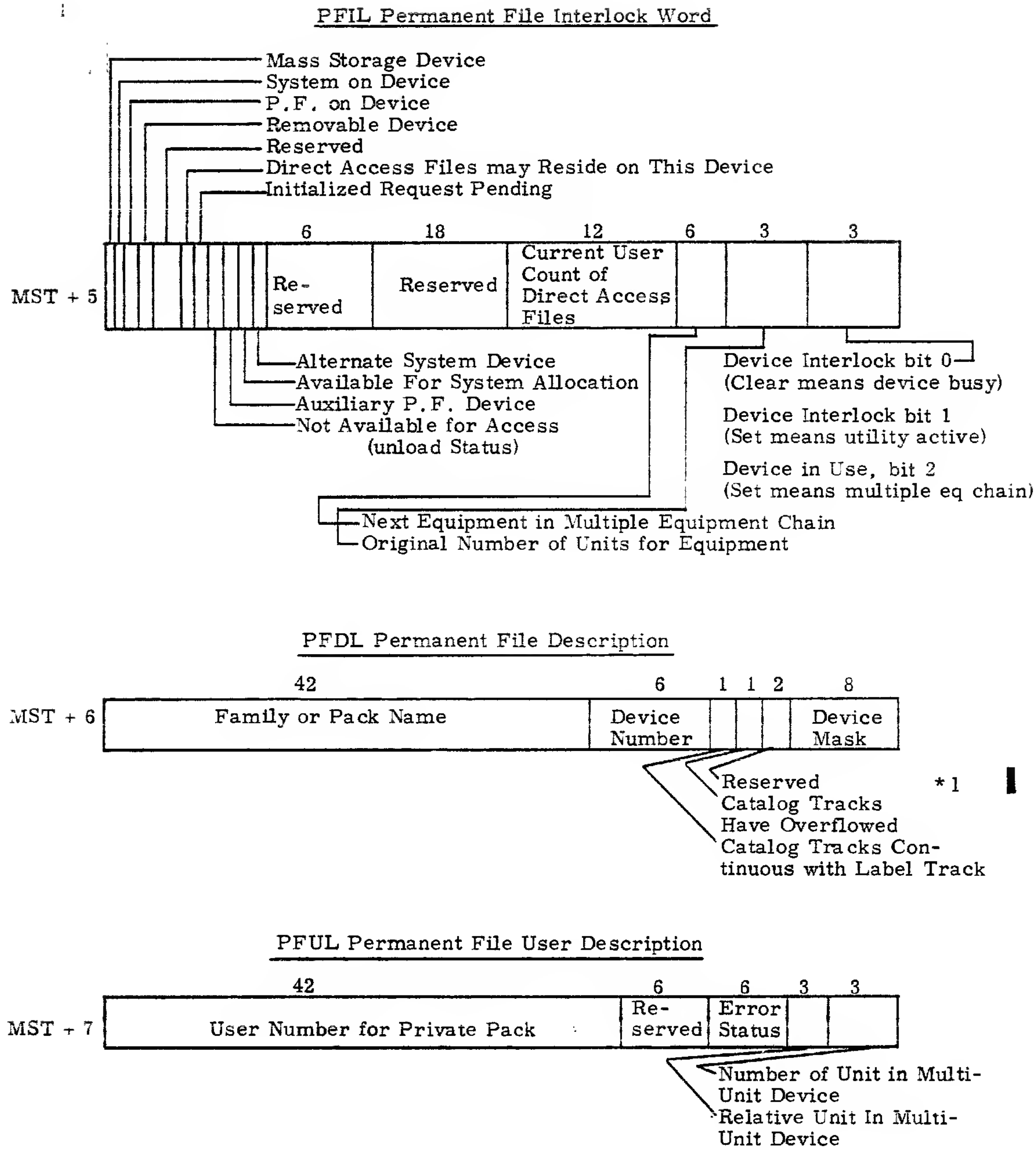


Figure 2-18. (MST) Mass Storage Table (Continued)

* 1 PFDL in MST

a. bit 11. Catalog track continuous with label.

b. bit 10. Catalog tracks have overflowed.

a. Normally the LABEL track is track 0 and on PF devices the catalog tracks will begin with track 1 and make a contiguous chain. In fact, the LABEL track is considered the first logical track in the label track chain and contains the system sector. It is possible due to flaws that track 0 is unavailable, in which case the label track will be the 1st available track in the catalog track chain.

If the tracks following the LABEL track are available, then they will be linked. The catalog tracks must be contiguous, however, they need not be contiguous to the last label track.

Commonly on a PF device that contains the SYSTEM, track 0 is the LABEL track and track 1 starts the file SYSTEM. The catalog tracks start after the system file. In this case, this bit is set on.

Note: LABEL track plus catalog tracks are considered one track chain, so the TRT will link these tracks together. This bit tells PFM where to look for the catalog tracks.

b. PFM computes the catalog track for a UI (see Install Handbook pp. IV-2-3 thru IV-2-8). If this track is full PFM overflows to other tracks (not any of the original catalog tracks). This bit just indicates to PFM that overflow may occur when searching for a hole for this operation. Also the 0 symbol is displayed on the EM display for the operator's edification.

Figure 2-18. (MST) Mass Storage Table (Continued)

Rest of MST (MSTL - 10)

Offset	Contents	
MST + 10	Reserved	
11	Reserved	
12	Reserved	not
13	Reserved	currently
14	Reserved	used
15	Installation Use	
16	Installation Use	
17	Installation Use	

Figure 2-18. (MST) Mass Storage Table (Continued)

Entry = 1 Byte + 1 Bit for each track

Track Link byte may take 1 of 2 forms

1.

1	11
1	Next Track In Track Chain

2.

1	11
0	EOI Sector in File

The next track is found from

1	9	2
1	Word in TRT	Byte in Word

TRT Word

12	12	12	12	4	4	4
Track Link	Track Link	Track Link	Track Link			

First Track of Direct Access P. F.
Track interlock bits
Track reservation bits

Figure 2-19. TRT Track Reservation Table

RCL - Resident CPU Library

(Absolute programs only)

	1	41	18
Type "OVL"	0	Program Name	Length

Length: Links to next program

	1	41	18
Type "ABS"	1	Program Name	Length

Length: Links to next program

RPL - Resident PPU Library

18	6	12	12	12
Program Name	0	Load Address	0	Length

Length: Links to next program

PLD - PPU Library Directory

18	6	12	12	12
Program Name		Load Address	Track	Sector

6-bit field: Alternate System Device EST ordinal if it exists

Figure 2-20. KRONOS Library Formats

Type "OVL"

	5	1	6	6	
Program Name	0	Scope Rec Flag*	Alt Dev Eq No	0	
12 Bits 0 | *1 0 24 Bits		Track 12 Bits * 2		12 Bits Sector* 2	* 3

Type "ABS"

Set by SYSEDIT because RFL= SEP was present.

42		5	1	6	6
1st Entry Point Name		0	Scope Rec Flag	Alt Dev Eq No	No. Entry Pts.
12 Bits FL Required	24 Bits *1 0	12 Bits Track *2		12 Bits Sector *2	
2nd Entry Point Name					
⋮					
NTH Entry Point Name					

Type "ULIB"

42	6	12
Library Name	1	
(36)	Random Address BIAS (24)	

42	6	12
Procedure Name	0	0
0 (36)	Random Address BIAS (24)	

The user library definition entry for a given compiler will exist after the (0, 0) overlay entry for the compiler.

ULD User Library Definition

Type "ULD"

1	41	18
0	Library Name	0

Type "COS"

T | Program Name			
		Track	Sector

T=0 → P Mode * If set, specifies that card cracking from TCS is done in SCOPE format. If not set, then TCS card cracking is in KRONOS format.

*1 Used to specify ALT system TRK and sector if specified.

T=1 → I Mode *2 Used to specify system TRK and sector.

Figure 2-21. CLD - CPU Library Directory

* 3 If alt dev eq no = 0 and * 1 $\neq$ 0, then CM resident routine and * 1 is address of routine relative to FWA of RCL. In fact, there is a CLD entry for every CP routine on the system, and 1AJ never needs to search the RCL explicitly.

Figure 2-21. CLD - CPU Library Directory (Continued)

The following is a copy of a dump of part of the CLD and a CATALOG of this section.

ABSOLUTE DUMP FROM 027510 TO 030677 PAGE

027510	-20323000000000000000	BCS	000000000000400--.J40	SC B	23312305041124000001	SYSEDIT
027513	05600000000040030074	E$ SC $	40040000000000000233	SD B0	24222242320000000000	TSS70
027516	00000000000040030147	SEA*	04251520241230000001	DUMPTK A	00500000000040070076	$ S9 *
027521	40040000000000000106	SD AF	01022305152000000000	ADSDMP	00000000000040070110	S0AH
027524	03100523230000000000	CHESS	00000000000040070140	S0AS	01200031604000000012	APPEND J
027527	00120000000040120035	J SJ Z	01242401031000000000	ATTACH	03102116070500000000	CHANCE
027532	04050611160500000000	DEFINE	07052400000000000000	GET	20010312150115000000	PACKNAM
027535	20052215112400000000	PERMIT	20252207050000000000	PURGE	22052014010305000000	REPLACE
027540	23012505000000000000	SAVE	40100000000000000000	SH	20060124033400000000	PFATCI
027543	00000000000040120046	SJ -	20060312434000000000	PFCAT1	00000000000040120070	SJ *
027546	20060317203134000000	PFCOPYI	00000000000040120130	SJAX	20060425152034000000	PFDUMPI
027551	00000000000040130000	SK	20061417010434000000	PFLOADI	00000000000040130052	SK)
027554	20062300000000000006	PFS F	00250000000040130144	U SKA9	20061417010400000000	PFLOAD
027557	20060425152000000000	PFDUMP	20060312400000000000	PFCAT	20060124030000000000	PFATC
027562	20060317203100000000	PFCOPY	40040000000000000537	SD E4	22252207011414000000	PURGALL
027565	00160000000040140015	N SL M	15231100000000000002	NSI 0	00540000000040140035	* SL Z
027570	06140127000000000000	FLAW	40040000000000003272	SD Z<	01030317231624000002	ACCOUNT 0
027573	00070000000040150071	G SM +	06011511143100000000	FAMILY	40160000000000000272	SN R<
027576	03100122070500000001	CHARGE A	00120000000040150077	J SM $	40550000000000000500	S* F
027601	11230400000000000001	ISF A	00220000000040150210	Z SMAH	40040000000000000746	SD 0-
027604	15170426011400000003	MODVAL C	01620000000040150121	AI SMAG	14111511242200000000	LIMITS
027607	20012323271722000000	PASSWOR	40140000000000012711	SL AWI	20221706111405000001	PROFILE A
027612	01150000000040160054	AM SN #	40440000000000002323	S9 $S	23052300000000000000	SFS
027615	00000000000040160132	SNAZ	05353330320000000000	EZ00CP	00000000000040170062	SO I
027620	04012401040506000000	DATADEF	00000000000040170132	SOAZ	04012401130120000000	DATAMAP
027623	00000000000040200122	SPAR	04020617221500000000	DSFORM	00000000000040210063	SO F.
027626	13242304152000000000	KTSDMP	00700000000040220056	* SR ,	14110224012313000000	LIBTASK
027631	01410000000040220145	A6 SRA+	20220523111500000000	PRESIM	00000000000040230071	SS +
027634	24220116053000000000	TRANEX	00000000000040240073	ST >	24220116053034000000	TRANEXI
027637	00000000000040250065	SU ,	24220116053050000000	TRANEX2	00000000000040250122	SUAZ
027642	24220116231115000001	TRANSIM A	02550000000040250141	B SUA6	40040000000000000122	SD AZ
027645	02041514110200010000	BDHLIB A	00000000000000004322	0R	24220116141102010000	TRANLIBA
027650	00000000000000004447	.90	23241115251401000000	STIMULA	00310000000040270010	(SW H
027653	24051405300000000000	TELEX	00000000000040270031	SW Y.	24051405303400000000	TELEXI
027656	00000000000040270074	SW $	24051405303500000000	TELEX2	00000000000040300003	SX C
027661	02140116130000000001	BLANK A	00160000000040300114	N SXAL	40440000000000001465	S9 L*
027664	15010716052400000000	MAGNET	00000000000040300131	SXAY	15010716052434000000	MAGNETI
027667	00000000000040300143	SXA0	01232311071600000010	ASSIGN M	01000000000040300147	A SXA*
027672	14010205140000000000	LABEL	22052125052324000000	REQUEST	22052317252203000000	RESOURC
027675	26231600000000000000	VSN	14061500000000000000	LFM	20051500000000000000	PFM
027700	22052100000000000000	REQ	40550000100000000232	S# M 0Z	06112215141300000000	FIRM66X
027703	00000000000040320027	SZ W	03012401141707000000	CATALOG	01050000000040320062	AE SZ)
027706	03012414112324000000	CATLIST	00320000000040320077	Z SZ $	03132000000000000002	CKP 0
027711	01000000000040320120	A SZAP	23062000000000000000	SFP	40240000000000000252	ST 0I
027714	03171515051624000014	COMMENT L	00040000000040320136	O SZA3	15170405000000000000	MODE
027717	16170530112400000000	NOEXIT	17160524011240000000	ONEXIT	17162327000000000000	ONSW
027722	17060623270000000000	OFFSW	22061400000000000000	RFL	22171414172524000000	ROLLOUT
027725	23052420220000000000	SETPR	23052424140000000000	SETTL	23251100000000000000	SUI
027730	23271124031000000000	SWITCH	25230503202500000000	USECPU	03011414000000000002	CALL 0
027733	00440000000040320141	9 SZA6	05301124000000000000	EXIT	07172417000000000000	GOTO
027736	04112320140131000003	DISPLAY C	00160000000040330001	N S0 A	11060000000000000000	IF
027741	23052400000000000000	SET	40400000000000000000	S5	03172031000000000004	COPY D
027744	01230000000040330012	AS S0 J	03172031020600000000	COPYBF	03172031022200000000	COPYBR
027747	03172031051100000000	COPYEI	03172031300000000000	COPYX	03172031230206000002	COPYSBF 0
027752	00520000000040330025	) S0 U	03172031030600000000	COPYCF	03172031032200000000	COPYCR
027755	03172031414200000000	COPY67	00560000000040330034	* S0 1	03172031424100000000	COPY76
027760	00560000000040330041	* S0 6	04150400000000000007	DMD 0	40510000000040330046	S(S0 -
027763	04152000000000000000	DMP	14020300000000000000	LBC	14170300000000000000	LOC

```
    CATALOG OF SYSTEM          FILE      1                                      74/07/21. 20.17.54.       PAGE      11
 C  NAME       TYPE          LENGTH   CKSUM     DATE        COMMENTS

195  90A       PP (6237)         71    2360   74/04/26. 73/05/24. 74/04/26. 026 - FILE COMMANDS.
196  90B       PP (6237)         66    3741   74/04/26. 73/05/24. 74/04/26. 026 - LINE ENTRY AND DATA MOVE.
197  90C       PP (6237)         62    0315   74/04/26. 73/05/24. 74/04/26. 026 - DISPLAY, TAB, DUP AND SCAN CONTROL.
198  90D       PP (6237)         50    1306   74/04/26. 73/05/24. 74/04/26. 026 - LINE SEARCH COMMANDS.
199  90E       PP (6237)         76    4244   74/04/26. 73/05/24. 74/04/26. 026 - RECORD SEARCH COMMANDS.
200  90F       PP (6237)         67    6361   74/04/26. 73/05/24. 74/04/26. 026 - REPLACE COMMANDS.
201  90G       PP (6237)         55    3507   74/04/26. 73/05/24. 74/04/26. 026 - MISC. COMMANDS.
202  AOC       PP (1100)       1137    0513   74/04/26. 71/01/09. 72/05/19. ANDY CAPP DISPLAY.
203  BAT       PP (1100)       1170    4076   74/04/26. 71/03/02. 73/05/08. BASEBALL GAME.
204  TLP       PP (1100)        146    1527   73/06/19.
205  RSE       PP (1100)         71    1441   73/06/19.
206  DPW       PP (1100)         44    5557   73/06/19.
207  TS570     OVL 00.00      25457    3131   73/06/19. TS570
208  DUMPTK    ABS             1131    6702   73/06/19.
       DUMPTK
       RFL=
       SSJ=
209  ABSDMP    OVL 00.00       1063    0566   73/06/21. ABSDMP
210  CHD       PP (1100)        754    1753   74/04/26. 71/01/09. 73/12/17. CHESS DISPLAY DRIVER.
211  CHESS     OVL 00.00      27076    2520    02/10/70.
212  DOG       PP (1100)        337    3760   74/04/26. 73/05/05.           SNOOPY WW1 FLYING ACE.
213  92A       PP (3105)        750    4017   74/04/26. 73/05/05.           ACE - DISPLAY DATA.
214  051       PP (1100)        635    7376   74/04/26. 71/01/09. 73/05/08. 6612/DD60 DISPLAY ALIGNMENT TEST.
215  GYR       COS             5145    1712
216  WRM       PP (1100)        463    2473   74/04/26. 71/01/09. 72/05/19. WORM(S) DISPLAY.
217  (00)             SUM =  104707

218  PFILES    ABS             1052    1074   74/04/26. 73/05/24. 74/03/11. PERMANENT FILE MANIPULATOR.
       APPEND
       ATTACH
       CHANGE
       DEFINE
       GET
       PACKNAM
       PERMIT
       PURGE
       REPLACE
       SAVE
       RFL=
       SUM=
219  PFATC1    OVL 01.00       2130    6636   74/04/26. 73/05/24. 73/08/15. PFATC1 - CATALOG PF ARCHIVE TAPE.
220  PFCAT1    OVL 01.00       3752    6634   74/04/26. 73/05/24. 74/04/26. PFCAT1 - CATALOG PERMANENT FILE DEVICE.
221  PFCOPY1   OVL 01.00       2400    0205   74/04/26. 73/05/24. 74/04/26. PFCOPY1 - COPY ARCHIVE FILE UTILITY.
222  PFDUMP1   OVL 01.00       6106    7415   74/04/26. 73/05/24. 74/04/26. PFDUMP1 - PERMANENT FILE DUMP.
223  PFLOAD1   OVL 01.00       7074    6004   74/04/26. 73/05/24. 74/04/26. PFLOAD1 - PERMANENT FILE LOAD.
224  PFS       ABS             2352    0503   74/04/26. 73/05/24. 74/04/26. PERMANENT FILE SUPERVISOR.
       PFS
       PFLOAD
       PFDUMP
       PFCAT
       PFATC
       PFCOPY
       RFL=
       SSJ=
225  PURGALL   ABS              640    7602   74/04/26. 73/05/05. 74/04/26. PURGE ALL PERMANENT FILES.
       PURGALL
```

NOTE

Order for JBC

SYOT
BCOT
EIOT
TXOT
MTOT

For Each Job Origin Type, There Exists A 10B Word Job Control Area

	12		12	12	12
Input File	Initial Queue Priority	Lower Queue Priority	Upper Queue Priority	Priority Age Interval	Current Interval Count
Rollout File	"	"	"	"	"
Output File	"	"	"	"	"
Service Control	Initial CPU Priority	CPU Time Slice (MS * 100B)	CM Time Slice (Sec)	0	
	Maximum No. Jobs Or Users	Maximum FL For Any Job	Maximum FL For All Jobs	0	
Permanent File Control	Limit/100B For Indirect File Size	Limit/100B For Number of Files In Catalog	Limit For Cumulative Size of Indirect Files		Reserved
System Event Tag	Reserved				
	0				

Figure 2-22. Job Control Area (JBC)

LOC	DESCRIPTION	NAMES	DEFINITION
	59 47 35 29 23 11 0		
RA	(unused) / T / P / Sense Switches / FORTRAN Sense Lights		T=Storage Move Flag P=Pause Flag, if set, program will halt until cleared by go from console.
RA+1	Package Name / *1 / Arguments		Used to communicate with CPUMTR.
2 . . 63		ARGR	Parameters from the program call card.
64	Name / No. of Parameters	PGNR, ACTR	Name of program called by Control Card.
65	*2 / Reserved / Address of next word available for loading	LWPR	
66	*3 / Reserved / Job Origin Type / Res-erved / First Word of ob-ject pro-gram	XJPR, JOPR	Job Status
67	Reserved / *4 / Reserved	LDRR	Loader Status
70 . . 77		CCDR	Image of control card currently being executed.
100	*5 / Name of alternate user library (ULIB) *7 / *6		
101			Actual first word of user program

*1 Bit 40 is auto recall bit.

*2 Bit 59 is set if Compare/Move Unit (CMU) is present.

*3 Bit 59 is set if CEJ/MEJ available.

*4 Bit 29 is set if Load has completed.

*5 Bit 58 is set if program called from DIS.
Bit 56 is set if no automatic field length reduction.

*6 Map flags for LINK:

0001	Statistics and errors	0004	Entry points
0002	Block assignments	0014	Cross-reference of entry points

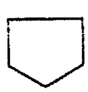
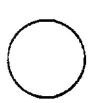
*7 If an overlay is loaded, ULIB is overlayed in bits 35-18 with lwa+1 of last largest overlay, i.e., the origin of the CM area that may be used for dynamic storage allocation.

Figure 2-23. Control Point RA Through RA + 100

MONITORS 3

3.0 ABBREVIATIONS AND DEFINITIONS

The following abbreviations and definitions are used throughout this and succeeding sections.

Abbreviation	Definition
CPUMTR	CPU monitor
MTR	PP monitor
EP	Exchange package
EPA	Exchange package area
CP	Control point
SCP	Sub control point
CPA	Control point area
CPU MTR or CP n	Physical central processor and hardware assigned to CPUMTR or CPn
PPn or PPn	Physical peripheral processor and hardware for PPn
E	CP Executive
MF	Monitor flag
EF	Error flag
MA	Monitor address of EP
EM	Error modes
( )	A symbol enclosed in parentheses means: contents of
	Flowchart continuation symbol for off page reference
	Flowchart start or continuation symbol for on page reference
MM	Monitor mode i.e., MF=1
PTX	Prior to exchange jump
ATX	After exchange jump
CPU	Central processor unit

Abbreviation	Definition
FWA	First word address
OR	PP output register
IR	PP input register
QP	Queue priority (priority that governs which jobs in the rollout and input queue gain access to control points), also which output queue entries gain access to printers
–	CPU priority (priority that governs which jobs at control points gain access to the CPU)
–	CPU Time Slice (the amount of time a job can use the CPU before it becomes a likely candidate for rollout)
–	CPU Time Slot (Job Switch Time) (the amount of time one control point can be active in the CPU prior to the CPU being given to another control point)
–	CM Time Slice (the amount of time a job can occupy central memory (control point) before it becomes a likely candidate for rollout)

3.1 CPU AND PP MONITORS

In KRONOS 2.1 there are two separate monitors: CPUMTR (central memory monitor) which controls CPU monitor mode execution and CPU scheduling, and MTR (peripheral processor monitor) which is in general control of the system and operates in PP0.

These two monitors work together, yet independently to allow the system to run smoothly and effectively.

Figure 3-1 is an overview of system interaction showing both monitors as a controlling entity. PPs communicate to CPU and vice versa through monitor by means of IR, OR, and RA+1 calls.

Figure 3-2 shows the interaction between this monitor concept and PP resident using the PP IR and OR.

Figure 3-3 shows the monitor interaction between CPU, PPU, and each monitor using the exchange jump feature. With the CEJ/MEJ option, the CPU program can either wait for PPMTR to call CPUMTR by finding (RA+1) ≠ 0, or the CPU program can directly call CPUMTR. PP routines may either wait for PPMTR to call CPUMTR by finding the (OR) ≠ 0, or call CPUMTR directly. Without the CEJ/MEJ option, CPU routines and PPU routines must wait for PPMTR to call CPUMTR for them.

Figure 3-4 shows the entry points for CPUMTR, while Tables 3-1, 3-2, and 3-3 show the monitor functions processed by CPUMTR.

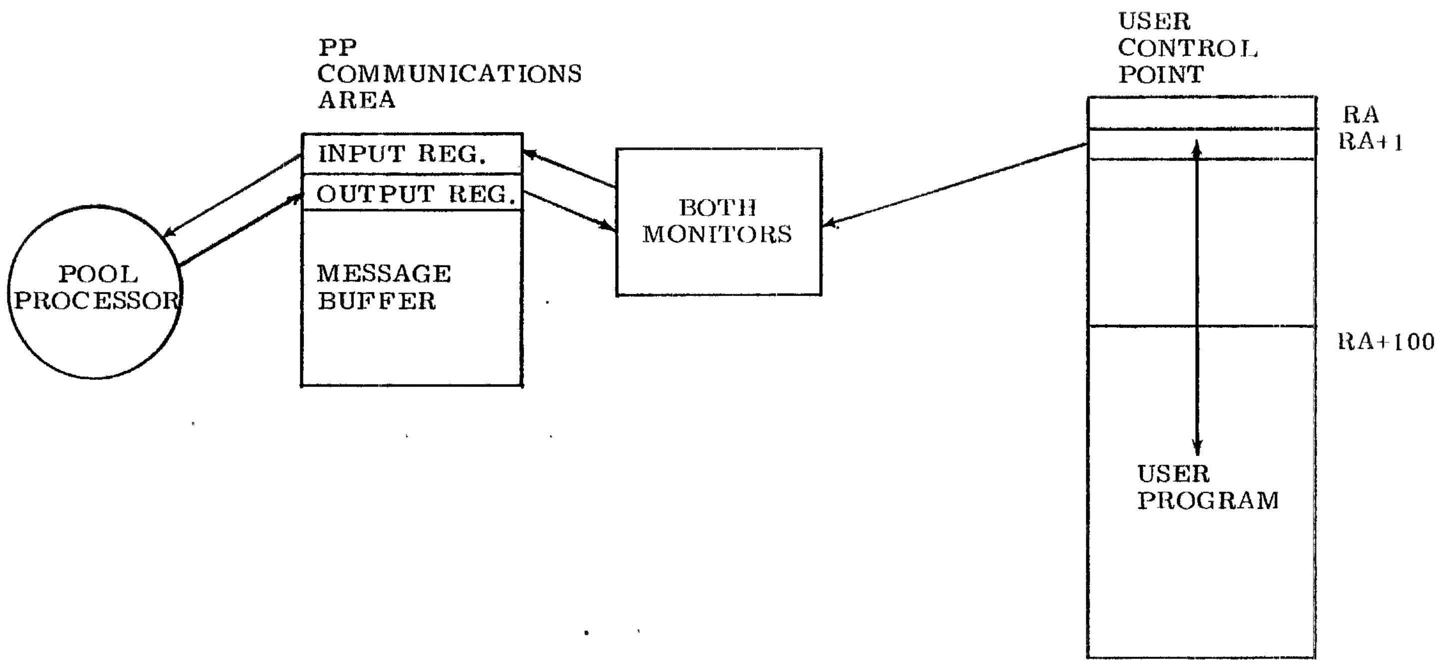


Figure 3-1. System Interaction

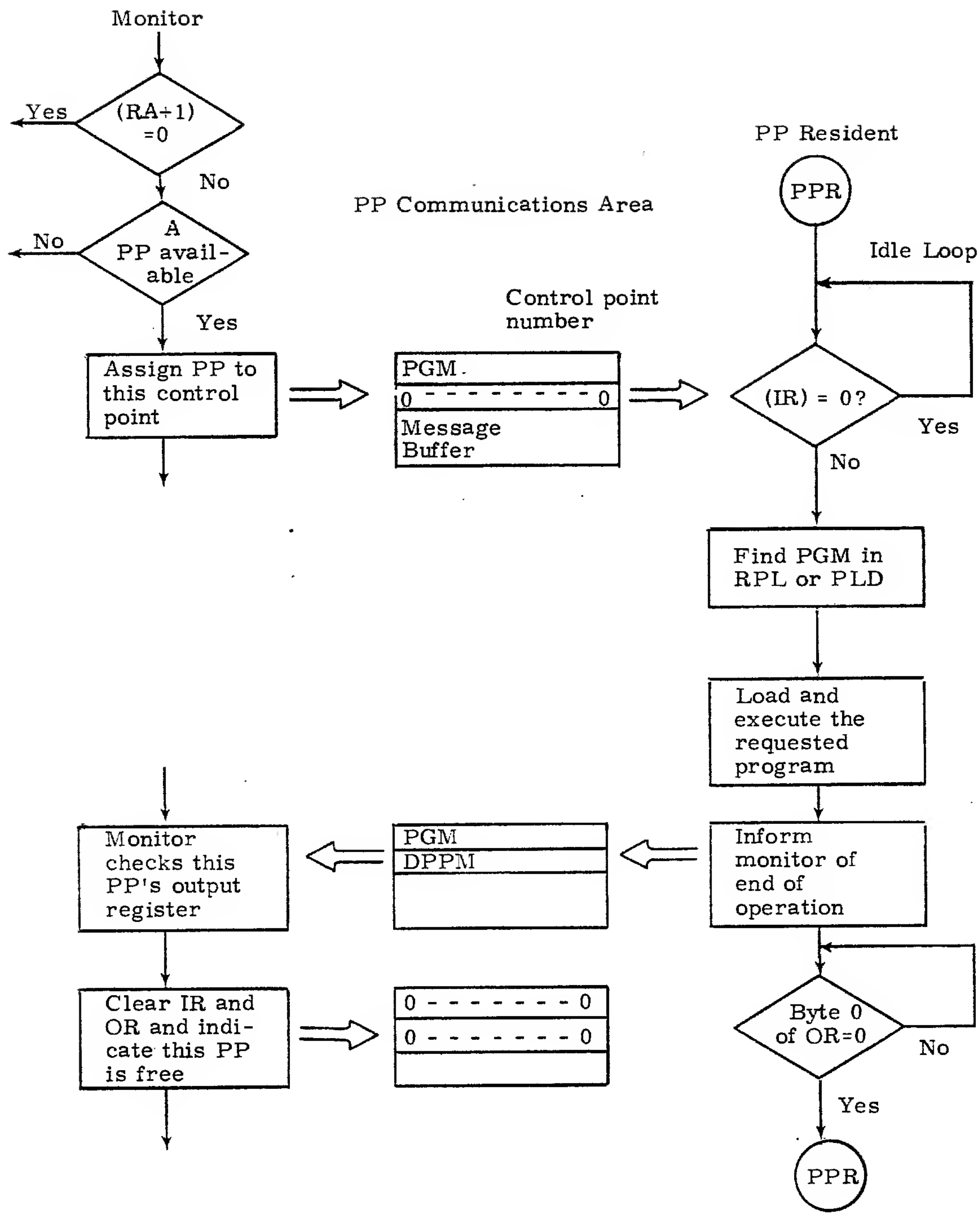


Figure 3-2. System Interaction

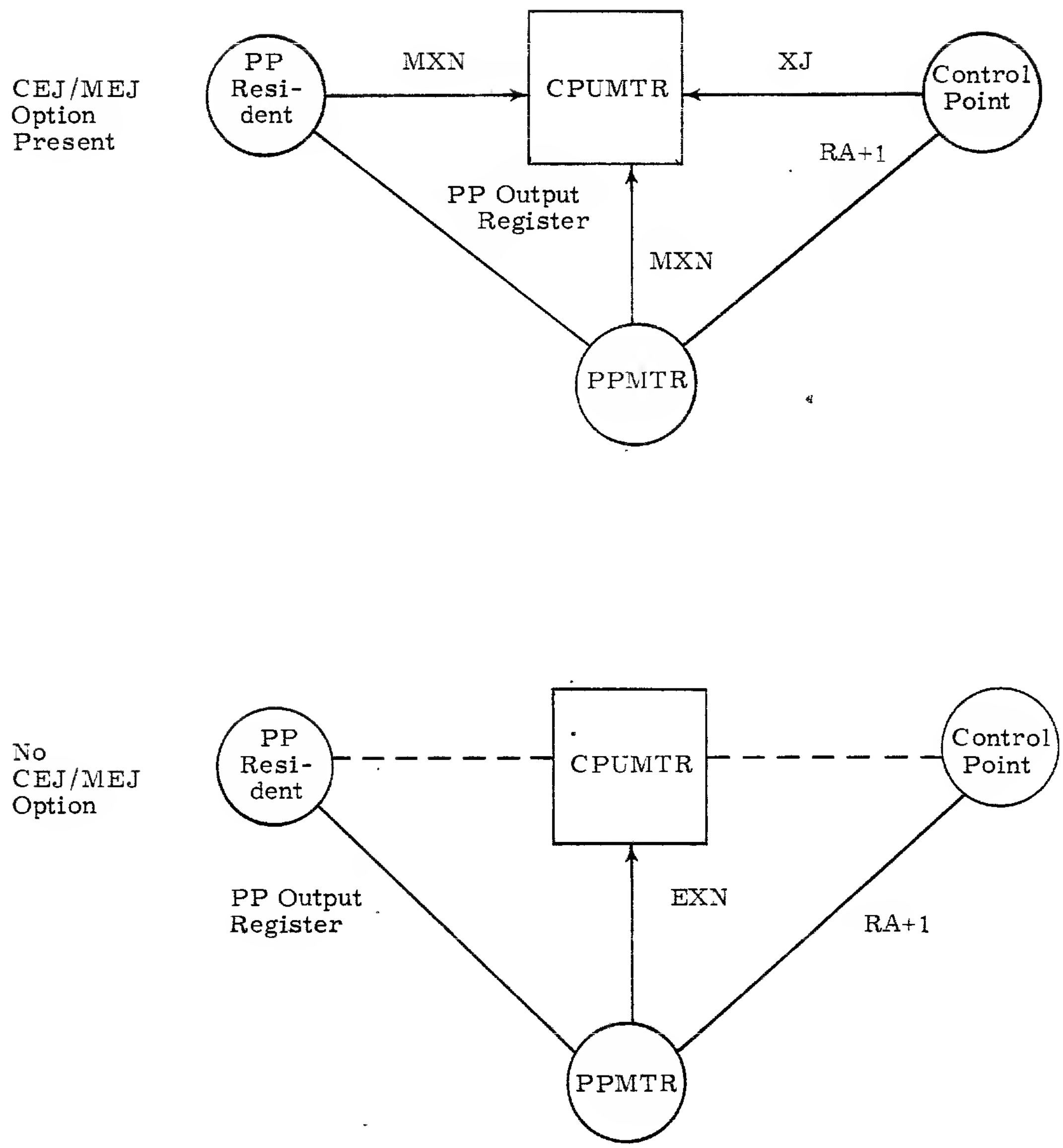
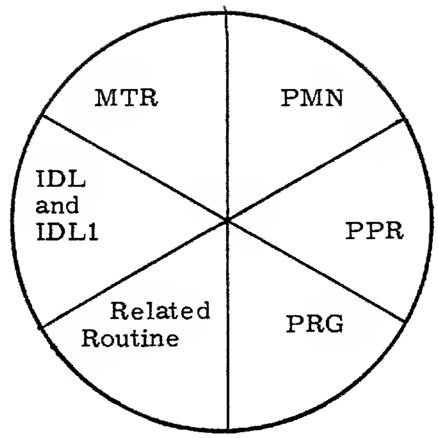


Figure 3-3. Monitors Interaction



Address in CPUMTR as of 8/1/73	Name	Description
20	MTR	From CPU program
717	PMN	From PPU monitor
1007	PPR	From pool PPU program
1357	PRG	Address where system CP begins execution in program mode. When system CP exchanges to the CPUMTR, CPUMTR will begin execution at MTR
1632 1635	IDL IDL1	From CPUMTR. These are idle loops for CP0 and CP1 respectively

Figure 3-4. CPUMTR Entry Points from Exchange Packages

All system interaction is affected using the exchange jump instructions.

The executable code of CPUMTR begins at location dayfile buffer + dayfile buffer length.

Functions processed by MTR for Pool PPs enter CPUMTR at PPR.*

TABLE 3-1. VALUES OF PP MONITOR FUNCTIONS (as of August 1973)

Name	Value	Description
AEQM	1	Assign equipment
AMSM	2	Assign mass storage space
CCHM	3	Check channel
DCHM	4	Drop channel
DEQM	5	Drop equipment
DFMM	6	Issue dayfile message
OFEM	7	Off equipment
ONEM	10	On equipment
PRLM	11	Pause for storage relocation
RCHM	12	Reserve channel
REMM	13	Request exit mode
REQM	14	Request equipment
ROCM	15	Rollout control point
RPRM	16	Request priority
RJSM	17	Request job sequence number
SCHM	20	Select mass storage channel
RSTM	21	Request storage
RSYM	22	Request system
SMSM	23	Set monitor step
STPM	24	Step monitor
TGPM	25	TELEX get pot
TSEM	26	TELEX request
DEPM	27	Disk error processor
DRCM	30	Driver recall CPU
SCPM	31	Select CPU(s) allowable for job execution
EATM	32	Enter Access system event table
	33-35	(Spares) unassigned

*The value determines that the function is intended for MTR.

Functions processed by CPUMTR, enter CPUMTR at PPR.

TABLE 3-2. VALUES OF CPUMTR FUNCTIONS (as of August 1973)

Name	Value	Description
ABTM	36	Abort control point
CCAM	37	Change CP assignment
CEFM	40	Change error flag
DCPM	41	Drop CPU
DJSM	42	Disable job scheduler
DTKM	43	Drop tracks
DPPM	44	Drop PP
ECSM	45	ECS transfer
RCLM	46	Recall CPU
RCPM	47	Request CPU
RDCM	50	Request data conversion
REWM	51	Read ECS word
RJAM	52	Request job accounting
RPPM	53	Request PP
RSJM	54	Request job scheduler
RTCM	55	Reserve track chain
SFBM	56	Set file busy
STBM	57	Set track bit
UADM	60	Update accounting and drop
WEWM	61	Write ECS word
JACM	62	Job advancement control
DLKM	63	Delink track chain
TDAM	64	Transfer data
TIOM	65	Tape I/O processor
RTLM	66	Request time limit
LCEM	67	Load central program from ECS
CSTM	70	Clear storage
CKSM	71	Checksum for reprieve
	72-75	(Spares) unassigned
MXFM	76	Maximum number of functions

Functions issued by MTR (only) and processed by CPUMTR enter CPUMTR at PMN.

TABLE 3-3. MTR FUNCTIONS PROCESSED BY CPUMTR

Name	Value	Description
ARTF	1	Advance Running Time
IARF	2	Initiate Auto Recall
MSTF	3	Move Storage
MRAF	4	Modify RA
MFLF	5	Modify FL
SCSF	6	Set CPU Status
SMSF	7	Set Monitor Step
CMSF	10	Clear Monitor Step
CAEF	11	Check Arithmetic Error
ACSF	12	Advance CPU Job Switch
PCXF	13	Process CPU Exchange Request

RA+1 REQUESTS PROCESSED BY CPUMTR

MSG	Send dayfile message		
CIO-CLO	CIO call		
ABT	Abort this CP		
LDR	Call absolute overlay LDR		
CPM	CP Functions	1)	Force upper
		2)	Set error exit
		3)	Read exit mode
		4)	Read Job Control word
		5)	Set job control word
		6)	Return user number
		7)	Read FL control word
END	End this CP normally		
RCL	Periodic or auto recall		
TIM	Request system time		
*RSB	Read subsystem control block		
RFL	Request field length		
XJP	Initiate Sub Control Point		
**TLX	Process special PPU request		
XJR	Process Exchange Jump request		
*SIC	Send Inter-CP block to subsystem		

*This request is only honored for jobs with "SSJ=" or **.

**This request is only honored for jobs whose queue priority is greater than MXPS.

NOTE: The format for the calls on these pages are contained in the Instant Manual and the EXT documentation of MTR and CPUMTR using the control card DOCMENT.

3.1.1 Monitor Function Descriptions (See Instant Manual for parameters)

3.1.1.0 MTR Functions

Function Number

1 AEQM - assign equipment

It is used by DSD/IDS for n. ASSIGN command. The equipment is reserved in the EST if it's not MS. Bits 47-52 of the EST will get the CP number to indicate the reservation.

2 AMSM - assign MS space

This function allows a PP to request n sectors of MS space on any TEMP device. See paragraph 7.5 p. 7-8 for further description.

3 CCHM - check channel

This function allows a PP to have a channel checked for availability. If the channel is free, it is assigned, if not free, the channel requested bit 11 in the CST is set. In any case control is returned to the PP immediately. Compare this with RCHM.

4 DCHM - drop channel

Simply sets assignment for this channel in the CST bits 7-10 to zero. It is used to release the channel reserved with RCHM or CCHM. This function is used by the PPR routine DCH, see chapter 4. This also does a Release unit reserve function when the device is MS and the R option is set for a dual access controller. See the CMRDECK MS EST entry in the Install Handbook.

5 DEQM - drop equipment

This function releases the equipment by setting bits 47-52 of the EST entry to zero. It is used to release equipment reserved with the AEQM, or REQM.

6 DFMM - process dayfile message

This function allows a PP to send a dayfile message to any of the system or CP dayfiles. This is used by the PPR routine DFM, see chapter 4.

7 OFEM - off equipment

This function sets the OFF/ON bit 23 in the EST on. Note bit 23 =

0 equipment ON

1 equipment OFF

10 ONEM - on equipment

This function sets the OFF/ON bit 23 in the EST OFF.

11 PRLM - pause for storage relocation

Any PP which determines that its CP has a storage move request pending (CMCL word 57 byte 0) must issue this function. MTR will not move the CP until all PP activity for that CP has ceased which is either a DPPM, PRLM, or CCAM, etc. This function is used by the PPR routine PRL see chapter 4.

12 RCHM - request channel

This function sets the CST bits 7-10 to the CP number, thereby assigning the channel, for whichever of up to 4 channels is available. The RCHM will not return control to the PP until the channel can be reserved. Compare with the CCHM which returns control whether the channel can be assigned or not.

13 REMM - request exit mode

This function sets the EM in the EP to the specified 12 bits. The EM register is in EP+3.

	3	6	3	12	18	18
EP+3	EM		EM		A3	B3

14 REQM - request equipment

This function allows the PP to request an equipment. Control is returned whether the equipment is available or not.

15 ROCM - rollout CP

This function sets the rollout requested bit (bit 24 in word JCIW, 22 of the CPA) on. A PP routine cannot force a job to rollout immediately, it must request rollout action. MTR will determine when the job should be rolled out and MTR will issue the JACM request option 5. See JACM.

16 RPRM - request priority

This function will set the CPU or queue priority in the CPA (word JCIW, 22).

17 RJSM - request job sequence number

This function returns the current job sequence number from JSNL word 22, and bumps it by one.

20 SCHM - select channel

Allows a PP to request the best channel for a multi-channel device. See paragraph 7.5 page 7-10.

21 RSTM - request storage

This function allows a PP routine to change the FL at a CP. The request is the amount of FL desired at the CP. If the request is for the same amount of FL or less already assigned at the CP, then the request is honored immediately. If the request is for an increase, storage moves may be necessary. Control is returned immediately in any case. If a PP wishes to reduce FL it should make this request. If it wishes to increase FL it should not make the request. It should place the FL increase required in FCLW word 60 byte 4. 1SJ will then schedule the increase on a priority basis. If a PP makes its own increase storage request with this function directly, it could seriously affect scheduling. The PP programmer should use the common routine COMPRSI to make increase storage requests.

22 RSYM - request system

This is the same as RCHM except that a system device is desired. See paragraph 7.5 p. 7-8. It is used to get a system device which is on the first available channel.

23 SMSM - set monitor step

This function is only accepted from DSD, any other PP will be hung. When the operator types in STEP or UNSTEP, DSD will issue this function. STEP mode forces MTR to accept only 1 function at a time under direction of DSD, see STPM. MTR will step CPUMTR and control the processing of those functions, see SMSF. DSD can specify to step system or only one CP. MTR will reissue all CPUMTR functions that were stepped when an unstep is issued from DSD.

24 STPM - step monitor

This function is only accepted from DSD, from any other PP it will be treated as a NOOP. (At a future data, this will result in a hang). When the operator hits the space bar, DSD will issue this request and then MTR will process one function, which may be to tell CPUMTR to process that function if it's a CPUMTR function.

25 TGPM - TELEX get POT

This is used to get a POT chain from TELEX. It is useful because the PP does not need to interrupt or start up TELEX for the request. See paragraph 13.3.2 monitor request queries.

26 TSEM - process TELEX request

Used to request various procedures from TELEX. See paragraph 13.3.2 monitor request queues.

27 DEPM - disk error processor

Used for MS error processing. See figure 7-10.

30 DRCM - driver recall CPU

Used to issue an RCLM if the CP is in periodic recall status. This function allows the PP to ask monitor to determine the CP status than do it itself. This request does not require an exchange jump, therefore the PP needs only to place the request in its OR and does not need to wait for it to be processed. This is critical for MS or tape drivers, who could loose a revolution or tape speed if it needed to wait for a CPUMTR request. However, the routine must wait for OR clear before again issuing this function. This is why MS error processors must wait for OR clear.

31 SCPM - select CPUs allowable for job execution.

It sets the JCIW word 22 of CPA byte 4 to:
- 0 any CPU
- 1 CPUD only
- 2 CPU1 only

32 EATM - enter-access system event table

Enter or read events to or from system event table. See paragraph 5.2.10.

3.1.1.1 CPUMTR Functions

36 ABTM - abort CP

Exactly that, abort the CP to which this PP is assigned. It sets PPET error flag and performs a DPPM.

37 CCAM - change CP assignment

Used to change the CP assignment for this PP. It reduces the PP count in the CP at STSW bits 52-48 in the OLD CP assignment, and increases it by 1 for the NEW CP assignment.

40 CEFM - change error flag

Replaces bits 36-47 in STSW word 20 of CPA. It's used to set or clear the error flag.

41 DCPM - drop CPU from CP

If CP is in "W" status it is placed in zero status. Note, since there is PP activity the CP will not be advanced.

42 DJSM - disable job scheduler

When a PP desires to do an activity which can affect the scheduler or in which the scheduler can affect it seriously, then it is desirable to disable 1SJ. For example, on the PURGE DSD command, 1DS will attempt to purge some file from a queue (input, output, or rollout). If 1SJ attempts to schedule that job while 1DS attempts to purge it, problems can occur.

When a PP issues the DJSM function it receives return status stating:

1. 1SJ is active
2. function accepted

In the case of 1, the PP needs to reissue the function.

In the case of 3, CPUMTR will note which PP issued the DJSM and will ignore all RSJM functions from other PPS. (They will not be hung but will be returned as if the function was accepted, i.e., RSJM will be treated as a NOOP). Multiple PP's can disable 1SJ at the same. However, all of them must issue an RSJM before 1SJ can be reenabled.

When the PP which issued the DJSM, finally issues an RSJM, then this RSJM and any others will be accepted and processed. Thus, one PP routine can disable 1SJ by DSJM and reenable it with the RSJM.

43 DTKM - drop tracks

This is executed in PM. Used to drop trailing tracks from a track chain. See paragraph 7.5 p. 7-8.

44 DPPM - drop PP

This is the last function issued before a PP jumps to its idle loop. It signifies that this PP routine is done and the PP is available for other assignments.

45 ECSM - ECS transfer

Used to get 101B words transferred from ECS to/from the ECS/PP BUFFER.

46 RCLM - recall CPU

Used to change the CP status from periodic recall to CPU candidate, i.e. "X" status to "W" status.

47 RCPM - request CPU

Used to start the CPU for this CP and set the CP status = "W". See STSW word 20 in CPA byte 0.

50 RDCM - request data conversion

Used to convert 30 bit integer to F10.3 display code format

51 REWM - read ECS word

Used to transfer one ECS word to the MB.

52 RJAM - request job accounting

Convert accounting information in CPA to F10.3 display code. Accounting information begins at ACTW and its length is an assembly constant. At level -6, the length is 5, so

words ACTW thru ACTWE, words 50 thru 54 in CPA are converted. It converts the lower 30 bits to F10.3 format for transfer to dayfile. 1CJ is the only routine using it. 1CJ must write this information on the users dayfile.

53 RPPM - request PPU

Used to start a PP routine in some other PPU. The response indicates whether the PP was assigned or none available. A PP can read PPAL and determine in advance if a PP is available. This will save time and overhead.

54 RSJM - request job scheduler

See DJSM and 6.1 1SJ p. 6-1 and JSCL word 40 or CMR. This function is used to interlock scheduler calls, so that only one copy of 1SJ is running at one time in the system.

55 RTCM - request track chain

This is executed in PM. This allows the PP routine to request a specified number of sectors and reserve the proper track chain.

56 SFBM - set file busy

Used to interlock the FNT/FST entry for a specific file. A PP will issue this function to reserve the file and when done releases the file itself by setting bit 0 of the FST to one. SFBM will set bit 0 of the FST to zero. This function can be used to interlock any word in CM, such as PFNL, or any word in the MST. If SFBM is issued for an FNT/FST, the filename word must also be provided to check that another PP has not dropped the file just after the PP issuing SFBM found it. Note, in both the FST and the FET if bit 0 is set, the file is NOT busy.

57 STBM - set track bit

This is executed in MM unless SYSTEM CP is active, then it's done in PM. Used to set the w, d, or i bits in the TRT. See paragraph 7.2 p. 7-3.

60 UADM - update accouting and drop

Used to interlock any counter in the CPA. The CPA word specified is incremented by one. If no word is specified the PP activity count in STSW is incremented by 1. This is the pseudo activity count at a CP. It is used mainly for tape jobs so the job cannot be completed or advanced, but it can be rolled out. See STSW figure 2-3.

61 WEWM - write ECS word

Used to transfer one word from the MB to ECS.

62 JACM - job advancement control

Options 1, 2, 3, 4 are used to set or clear the job advancement flag at a CP with implied DPPM if desired. PP routines should not call 1AJ directly for job advancement. MTR will decide when a job needs to be advanced and will issue JACM option 5 to call 1AJ to the job. 1AJ then decides if the CP needs advancement or rollout.

63 DLKM - delink tracks

This is executed in PM. DLKM is used to drop intervening tracks on an existing file chain and relink the file chain properly. An excellent example would be PFM delinking his indirect (IPF) file chain in response to some user issuing a PURGE on some IPF which is long enough to completely cover several tracks. PFM attempts to keep his IPF file chain to a minimum size when possible. The CPUMTR DOCUMENT description is as follows (correction ident CI = CPUMTR 2974 level 4):

```
    DLKM - DELINK TRACKS.

    ENTRY
OR  12/ DLKM,12/ EQ,12/ FT,12/ NT,12/ LT
    EQ      EQUIPMENT NUMBER
    FT      TRACK ONTO WHICH NT IS LINKED.
    NT      TRACK TO BE LINKED TO FT.
    LT      LAST TRACK IN CHAIN TO DROP.

    BIT 11 OF FT MUST BE CLEAR
    ALL TRACKS FROM FT (NOT INCLUDING FT) TO LT ARE RELEASED
    NT IS LINKED TO FT.
```

The instant manual description is:

63 DELINK TRACKS DLKM

REQUEST: OR 0063 00 eq ffff nnnn 1111

CPUMTR terms - Instant terms		
	eq	Equipment number
FT=ffff	ffff	Track onto which nnnn is linked (bit 11 of ffff must be clear)
NT=nnnn	nnnn	Track to be linked to ffff
LT=1111	1111	Last track in chain to drop

REPLY: OR 0000 0000 0000 0000 0000

NOTE: DLKM will drop all tracks starting at the track linked to by FT and ending and including track LT. Track FT will be linked to track NT. If track LT did not link to track NT previously, we have a serious condition. See example b.

a. As an example let's say we have a track chain of 7 tracks linked in the following manner. 12 → 4 → 10 → 15 → 22 → 11 → 20 with EOI in sector 16. Then the TRT will be:

0	1	2	3	
4 4010	5	6	7	
10 4015	11 4020	12 4004	13	
14	15 4022	16	17	
20 0016	21	22 4011	23	

Suppose we wish to delink this chain by eliminating tracks 10, 15, 22 and 11 and relink track 4 → 20. Hence, BEFORE:

```
12 → 4 → 10 → 15 → 22 → 11 → 20
     ↑                  ↑    ↑
    ffff               1111 nnnn
     FT                 LT   NT

OR= DLKM   00eq    ffff   nnnn   1111
    0063   00 eq   0004   0020   0011
```

AFTER: 12 → 4 → 20

The TRT will be:

0	1	2	3	
4 4020	5	6	7	
10	11	12 4004	13	
14	15	16	17	
20 0016	21	22	23	

b. Suppose we specified the call incorrectly

OR = 0063 00eq 0004 0020 0022

then we have the following problem:

```
BEFORE:  12——>4——>10——>15——>22——>11——>20
               ↑              ↑          ↑
              ffff           1111       nnnn
```

AFTER: 12——>4——>20 but also 11——>20

and track 11 is not part of this chain or any other chain. There is obviously a problem and at this time it is not known whether CPUMTR will diagnose this problem.

64 TDAM - transfer data to/from job - from/to MB

Allows a PP to transfer up to 6 words from/to MB - from/to a job. The address to transfer to/from is a relative address. The transfer must be to/from a subsystem. It alleviates the problem of a PP finding the subsystem and deciding if it is ready for reception of data. This is equivalent to the SIC/RSB facility except no inter CP communication area is necessary. See paragraph 5.3.6. p. 5-44. The real problem is being at one CP, and needing to write data at another without it being moved during the write.

65 TIOM - tape I/O processor

This function updates the tape accounting information, i.e. number of blocks transferred in MTUW word 53 of the CPA. Exit from this function is to CCAM to change the PP assignment to MAGNETs CP. If the completion code is non-zero, the specified UDT word is cleared, the FET is set complete, and the tape activity count is decremented in STSW word 20 byte 2. 1MT uses this function when it completes a read/write request on a tape. Since the UDT and the FET must be changed, and they are in two different CPs, this function prevents any problem by keeping the CP & MAGNET from interferring with each other. UDT must be cleared before FET is set complete or an I/O sequence error could occur. The problem again is storage move at one CP while attempting to write to it.

66 RTLM - request CPU time limit

Used to change the CPU time limit in CTLW word 20, byte 2, 3, and 4 in the CPA. The time limit exceeded flag in ACTW word 50 byte 0 is cleared.

67 LCEM - load central program

This is executed in PM. Used to load an ECS or CM resident routine into the CPs FL.

70 CSTM - clear storage

Used to clear a specified contiguious amount of CM. Memory is cleared backwards, i.e. address is LWA to clear.

71 CKSM - checksum a specified area

Checksum area from FWA to LWA+1 and compare to checksum in MB.

74 MXFN - maximum function number

This is used by a PP when it desires to hang itself for some reason it considers catastrophic. CPUMTR will see that it is out of range and will hang the PP.

NOTES on hanging PPs:

A PP is hung when one of the monitors determines that a function is illegal. e.g. function out of range, RCHM on some non existant channel, etc. If CPUMTR hangs a PP the message "PP HUNG" is displayed at the system CP.

If MTR hangs a PP the message is "HUNG PP".

In any case the packed date and time of the hang is placed in MB+5.

3.1.1.2 MTR functions to CPUMTR.

These are special functions and the request is transmitted via the X0 register instead of MTRs OR.

FUNCTION NUMBER

0 no name

entry (X0) = | 0 |

exit none.

This function tells CPUMTR that some CP has an RA+1 request. This is used for systems where the XJ is not available or the user's program is not doing an XJ.

1 advance running times

42	18
0	ARTF

entry (X0) = (42-bit field 0, 18-bit field ARTF)

exit none.

Update running times. Updates RTCL in CMR and ACTW in CPA and set time limit exceeded flag if time limit has been exceeded. It also checks for P = 0 and program stop. MTR checks active CPs and if the P does not change, MTR looks at the instruction P points to.

If the top parcel (top 15 bits) is zero, it is a PS and MTR calls CPUMTR with this function to register the error. If the PS is not in the 1st parcel, then the CP will not be interrupted by MTR and it will stop only on time limit error or operator drop.

2 IARF initiate auto recall

	24	18	18
entry (X0) =	0	CPA FWA	IARF

exit none.

MTR while in the routine PPL, process PP recalls, will check RA+1 of a CP in auto recall and if RA+1 set with auto recall requested, it will reissue the PP request. See PPL in chapter 3.

If a PP routine finds that it cannot process the request it was called for at this time, it can copy its IR back to RA+1 if the CP is in "R" status. When MTR goes thru its PPL routine it will find the request and have CPUMTR reissue it to a PP.

3 MSTF move storage

	42	18
entry (X0) =	0	MSTF

	12	30	18
(SMRR) =	IN/100	0	CPA FWA

where IN = + or - number of words to move the CP

exit (SMRR) =	0

This function asks CPUMTR to move a CPs entire FL up or down in CM the specified number of words. CPUMTR can get the original RA from the STSW word. SMRR is a local word in CPUMTR.

4 MRAF modify RA

	12	12	18	18
entry (X0) =	IN/100	0	CPA FWA	MRAF

where IN = + or - value to change RA.

exit none.

CPUMTR will change RA in STSW and EP by the specified amount.

5 MFLF modify FL

	12	12	18	18
entry (X0) =	IN/100	0	CPA FWA	MFLF

exit none.

CPUMTR will change FL in STSW and EP by the specified amount.

6 SCSF set (restore) CPU status

entry (X0) =

12	12	18	
STATUS	0	CPA FWA	SCSF

exit none.

CPUMTR will place the specified status in the STSW word. This is used when MTR issues the DCPM function. The status is returned to MTR for safe keeping. When MTR is ready to restart the CPU it will issue this function restoring the former status.

Functions MSTF, MRAF, and SCSF may all be used when a CP needs to have its FL changed via the RSTM function. If MTR has to move the CP, it will issue the DCPM and save the status, then issue the MSTF for the move. Note that MSTF will update the RA and FL. If no storage move is required, then the MRAF and/or MSTF will be used.

Finally, it issues SCSF to restore the former status. Note, when a CP is going to be moved, the only criterion for that move is no PP activity, so the CP could be in any status when MTR is ready to make the move, and after the move, the proper status must be restored.

7 SMSF set monitor step

entry (X0) =

42	18
0	SMSF

exit none.

This tells CPUMTR to disable his automatic processing of monitor functions and to wait for MTR to indicate which function to process. Interaction is accomplished via the DXP and DXJ stuff in CPUMTR. See figure 3-7. SMSF and CMSF are used when the system is placed in STEP mode. See SMSM and STPM.

10 CMSF clear monitor step

entry (X0) =

42	18
0	CMSF

exit none.

Re-enable automatic processing of monitor functions.

11 CAEF check arithmetic error

entry (X0) =

24	18	18
0	CPA FWA	CAEF

exit none.

CPUMTR will check (P), if zero, it gets the error flag in STSW to ARET=2, arithmetic error.

12 ACSM advance CPU job switch

	24	18	18
entry (X0) =	0	CPA FWA	ACSF

exit none.

Used to change the CP assignment of the CPU. It is used in the MTR routine JSW, process CPU job switching, to exchange the CPU from one CP to another, which is slot time exceeded processing.

13 PCXF process CPU exchange request

	42	18
entry (X0) =	0	PCXF

exit none.

If CPUMTR is executing in one of the CPUs and needs to be in the other CPU it will tell MTR of its plight via some interaction word and XJ. MTR will then issue this request to the other CPU. This is done in the AVC advance clock routine, which is the one section of MTR that must execute at least every 4 milliseconds. For example, ABTM. PPR doesn't know which CPU its CP is in, so it starts CPUMTR up in CPUO. If the CP to be aborted is in CPU1, then CPUMTR must get itself into CPU1 and that CP out of CPU1.

MTR processes Pool PP OR requests as follows.

If the CEJ/MEJ is not available or disabled, then MTR will check all OR requests. If a request is for CPUMTR, MTR will jump to its routine CPR. CPR will exchange in CPUMTR for that PP.

If the CEJ/MEJ is available, MTR will ignore any CPUMTR request, since the PP must issue its own MXN, i.e., CPUO cannot stop CPU1, so the PCXF alternate exchange request is made.

3.2 EXCHANGE JUMPS

An installation may make use of the optional hardware instructions MXN (monitor exchange) and XJ (exchange jump) or EXN (exchange). KRONOS 2.1 requires either the combination of MXN/XJ or EXN.

Exchange jumps use an exchange package as shown in Figure 3-5. A general description of this package is contained in Section 2.

	59	53	35	17 0	
n		P	A0	B0	
n+1		RA_{CM}	A1	B1	
n+2		FL_{CM}	A2	B2	
n+3	EM		A3	B3	
n+4	RA_{ECS}		A4	B4	
n+5	FL_{ECS}		A5	B5	
n+6		MA	A6	B6	
n+7			A7	B7	
n+10	X0				8
n+11	X1				9
n+12	X2				10
n+13	X3				11
n+14	X4				12
n+15	X5				13
n+16	X6				14
n+17	X7				15

Figure 3-5. Exchange Package

3.2.1 Control Data 6400/6500 Systems Central Processor Monitor

In Control Data 6400/6500 computer systems, system functions are normally handled by the monitor located in a peripheral processor. The 6400/6500 computer systems are equipped with certain hardware capabilities to effectively implement monitor activities in the central processor. Since the central processor can reference extended core storage directly for service routines, programs, and data, a central processor monitor program to handle these and other functions is faster and more efficient than a monitor residing in a peripheral processor.

The hardware elements of the 6400/6500 system which provide the essential capabilities for implementing a central processor monitor are described in the ensuing paragraphs.

3.2.2 Monitor Address Register (MA)

Contained in the exchange jump package (bits 36-53 of location "n+6") is an 18-bit monitor address. Just as other central processor operational registers are loaded during an exchange operation, so is the monitor address register loaded with the 18-bit monitor address. This monitor address is the starting address of the exchange package for an ensuing central exchange jump instruction (except when the monitor flag bit is set; refer to the instruction description).

3.2.3 Monitor Flag Bit

The Central processor has, in the central memory control section of the system, a monitor flag bit. A master clear (dead start) clears the monitor flag bit. Any action thereafter on this bit is via the monitor exchange or the central exchange jump instructions. (There is no instruction with which to sample the status of this bit directly and/or independently of these instructions.) The operation of this monitor flag bit is described under the instruction descriptions.

Mode	Flag Bit	CPU
Monitor Mode	1	Not interruptable
Program Mode	0	Interruptable

3.2.4 Central and Monitor Exchange Jump Instructions

With the CEJ/MEJ option two instructions exist for central processor monitor implementation. The first, XJ, executable by the central processor and the second, MXN, executable by the peripheral processors. These instructions are as detailed below.

3.2.4.1 Central Processor

code	mnemonic	description
013	XJ B_j+k	Central Exchange Jump (60 bits)

CPU Memory Layout

code	j	K	not used
59 51	50 48	47 30	29 0

This instruction unconditionally exchange jumps the central processor, regardless of the state of the monitor flag bit. The instruction action differs, however, depending on whether the monitor flag is set or clear. Operation is as follows:

- Monitor Flag bit clear

  The starting address for the exchange is taken from the 18-bit monitor address register. Note that this starting address is an absolute address. During the exchange, the monitor flag bit is set.

- Monitor Flag bit set

  The starting address for the exchange is the 18-bit result formed by adding K to the contents of register Bj. Note that this starting address is an absolute address. During the exchange, the monitor flag bit is cleared.

3.2.4.2 Peripheral Processors

code	mnemonic	description	
261	MXN d	Monitor Exchange Jump	(12 bits)

PP Memory Instruction Layout

Code	d
11 … 1	0

$$d = \begin{cases} 0 \text{ exchange to CPU0} \\ 1 \text{ exchange to CPU1} \end{cases}$$

This instruction, typically used to initiate central processor monitor activity, is a conditional exchange jump to the central processor. If the monitor flag bit is clear, this instruction sets the flag and initiates the exchange. If the monitor flag bit is set, this instruction acts as a pass instruction. The starting address for this exchange is the 18-bit address held in the peripheral processor A register. (The peripheral processor program must have loaded A with an appropriate address prior to executing this instruction.) Note that this starting address is an absolute address.

3.2.4.3 EXN

In an installation without the MXN/XJ instruction set, the EXN is the only exchange instruction available. It is a PP initiated exchange jump which occurs independently of the mode of the CPU and has no effect on the CPU mode. MTR is the only PP program that may perform an EXN; it must simulate the MXN for all PPs in the system and simulate XJ for the central processor. When MTR detects a request for CPUMTR in a PP output register, it will EXN to the exchange package for the pool PP which desires the excahnge jump.

code	mnemonic	description
260	EXN	Normal Exchange Jump

NOTE

PP memory instruction layout is the same as MXN.

3.2.5 Programming Notes

The following should be considered:

1) Note that any exchange (260, 261 or 013) to the exchange package will load the contents of location "n+6" into the monitor address register (other operational registers are similarly loaded). Thus, any ensuing 013 instruction using the contents of the monitor address register as a starting address uses those contents as loaded.

2) The exchange packages for entering the central processor monitor should usually have the Reference Address (RA) equal to 000000 and the Field Length (FL) equal to central memory size.

3) Since the monitor flag bit cannot be directly sampled, a program cannot directly determine its state; hence, success in performing a peripheral processor monitor exchange cannot readily be predicted. Further, program control always is given to the next instruction, whether or not the exchange is honored. A method of determining whether the monitor exchange occurred is as follows:

Table 3-4 summarizes the operational differences between the normal exchange jump instruction (260) and the monitor and central exchange jumps (261 and 013).

TABLE 3-4. EXCHANGE INSTRUCTION DIFFERENCES

	Instruction	Conditional/ Unconditional	Operational Differences: Effect on Monitor Flag Bit	Operational Differences: FWA of Exchange Package in CM
No CEJ/MEJ	EXN 260 (Normal Peripheral Processor Exchange Jump)	Unconditional	No effect on flag	Peripheral Processor A Register
With CEJ/MEJ	MXN 261 (Peripheral Processor Monitor Exchange Jump) MXN	Conditional (occurs only if Monitor Flag bit is clear; passes if flag is set)	Sets flag	Peripheral Processor A Register
	XJ 013 (Central Exchange Jump) with Monitor Flag Bit clear	Unconditional	Sets flag	Central Processor Monitor Address Register
	XJ K+(B_j) 013 (Central Exchange Jump) with Monitor Flag Bit set	Unconditional	Clears flag	Address formed by K+(B_j)

a) Set B0 (bits 0-17 of location "n") in the exchange package to 7777.

b) Initiate the monitor exchange (261).

c) Read B0 from the exchange package in central memory. If the monitor exchange was honored, B0 in the exchange package will equal 000000. If the instruction passed, this location still holds 7777.

4) Different exchange packages should be used for central processor exchanges and peripheral processor exchanges. This aids software determination of which of two jumps (central or monitor exchange jumps) was executed when both were initiated at approximately the same time.

5) Simultaneous exchange requests are resolved in favor of the central processor.

6) If either a 260 or 261 instruction is waiting to be honored when the central processor issues a 013_jk instruction, the 013 instruction is not executed and the peripheral processor exchange occurs. When control is returned to the exchanged program (the interrupted program containing the 013_jk instruction), the 013_jk instruction is re-issued and executed.

7) The state of the monitor flag bit has no effect on the operation of the normal PP exchange jump (260); nor has this instruction any effect on the flag.

In addition, there may be CPUMTR requests which require more CPU time than it is feasible for CPUMTR to use in monitor mode and still ensure smooth system flow. For these requests, such as DTKM (drop tracks), the CPUMTR will queue them at the system control point and exchange jump to this control point. The system CP operates in program mode and is treated as any other user program. If the system CP is interrupted with another long request, the request is placed in the system CP queue and the system CP is restarted. The system control point can be interrupted by any MXN from a PP. However, because its CPU priority is the highest in the system (100), it will always get the CPU back immediately. No other control point will get the CPU if the system control points wants it.

Figure 3-6 shows all the system exchange packages and the entry points into CPUMTR. Table 3-5 shows the correspondence between CP, CP address, and the exchange package MA for a system configured to have 17B control points

Note that each PPU has its own exchange package in CMR. The system CP and each normal CP has its own exchange package in CMR in the control point area.

A CP will always have (MA)= its exchange package address. Additional exchange packages are provided for the two idle routines, subcontrol points, disabled central exchange, return package, disabled central exchange program, and a simulated exchange exit to monitor mode. These packages are generated at the end of the CPUMTR code.

Note that PP0, MTR's exchange package, is not contiguous with the other PP exchange packages.

Figures 3-7 and 3-8 show the generation of these EPs in the CPUMTR listing. Note that if the machine has only one CPU, only one idle package is built. If the machine has a CEJ/MEJ option enabled, the DXP, DXJ, and SXJ package is not assembled.

3.3 FLOW OF EXCHANGES

The flow of exchanges (there are only four distinct types) are illustrated and explained in flow diagrams Figure 3-9 through Figure 3-12.

TABLE 3-5. CORRESPONDENCE BETWEEN CP, CP/ADDRESS, AND THE EXCHANGE PACKAGE MA IN A SYSTEM CONFIGURED TO HAVE 17B CPs

Control Point	Address	Xchg Pkg MA
1	200	200
2	400	400
3	600	600
4	1000	1000
5	1200	1200
6	1400	1400
7	1600	1600
10	2000	2000
11	2200	2200
12	2400	2400
13	2600	2600
14	3000	3000
15	3200	3200
16	3400	3400
17	3600	3600
20 (SYSTEM)	4000	4000

SYSTEM EXCHANGE PACKAGES

	PPUs*2	PPMTR	Control Point N+1	Control Points 1 ↔ N	Sub-control Points and Idle Programs
Graphic representation	PXP PPU (PP1) Exchange Package PPU(PPn) Exchange Package	MXP PPMTR Exchange Package (PP0)	SXP System Control Point (n+1) Exchange Package	200B Control Point 1 Exchange Package N*200B Control Point n Exchange Package	SCX Sub CP EP1 SCX1 Sub CP EP2 IXP IDLE CPU0 IXP1 IDLE CPU1
Significant Contents	P=PPR MA=zero B2=address of PP1 EP (PXP+(i-1) *21B)	P=PMN MA=zero B2=MXP	P=PRG MA=System Control Point Area Address =SXP	P=CP Prog P address MA= This Control Point Area Address =addr. of CP_i XJPKG [i*200B]	Sub CP P=MTR MA=SCX, SCX1 B2=SCX, SCX1 IDLE P=idle Loop Address (IDL, IDL1) See 3-58 MA=IXP, IXP1
Size, Numbers and Location	21 words per package. Up to 19 packages. These start at end of CPUMTR code *1	20 words for this package. This is at the end of CPUMTR	First 20 of system control point area in CMR	First 20 words of each control point area in CMR	20 words for each package. These are at end of CPUMTR.
Symbolic address	CPUMTR address PXP	CPUMTR address MXP	CPUMTR address SXP	200B 400B ⋮ N*200B	CPUMTR address SCX and IXP SCX1 IXP1

*1 The ~~21B words spaces~~ the packages so that no bank conflicts will arise when PPs access them on 65K systems.

*2 PRS of CPUMTR will dynamically set up either 9 or 19 packages at D/S depending on the hardware.

Figure 3-6. System Exchange Packages

```
                               QUAL  MONITOR                                        CPUMTR    4904
                       **      MXP - PPU MONITOR EXCHANGE PACKAGE, PP0              CPUMTR    4906
                                                                                    CPUMTR    4907
                                                                                    CPUMTR    4908
1763                   MXP     EXP - P=PMN,FL=(,MCM),FLX=(,MED,B1=1,B2=MXP          CPUMTR    4909
EXP is a macro which
generates an exchange
package.               **      SCX - SUB-CONTROL POINT EXCHANGE PACKAGES.           CPUMTR    4911
                                                                                    CPUMTR    4912
                                                                                    CPUMTR    4913
2003                   SCX     EXP P=MTR,FL=(,MCM),FLX=(,MEC),B1=1,B2=SCX           CPUMTR    4914
                                                                                    CPUMTR    4915
2023                   SCX1    EXP P=MTR,FL=(,MCM),FLX=(,MEC),B1=1,B2=SCX1,A0=1     CPUMTR    4916
                       **      IXP - IDLE EXCHANGE PACKAGES.                        CPUMTR    4918
                                                                                    CPUMTR    4919
                                                                                    CPUMTR    4920
2043                   IXP     EXP P=2,RA=/PROGRAM/IOL,FL=3,MA=IXP                  CPUMTR    4921
                                                                                    CPUMTR    4922
2063                   IXP1    EXP P=2,RA=/PROGRAM/IDL1,FL=3,MA=IXP1                CPUMTR    4923
                       **      DXP - DISABLED CENTRAL EXCHANGE RETURN PACKAGE.      CPUMTR    4925
                                                                                    CPUMTR    4926
                                                                                    CPUMTR    4927
2103                   DXP     EXP P=DXJ+1,FL=(,MCM),X0=1                           CPUMTR    4928
                       **      DXJ - DISABLED CENTRA L EXCHANGE PROGRAM.            CPUMTR    4930
                                                                                    CPUMTR    4931
                                                                                    CPUMTR    4932
2123  0130002103 +     DXJ     XJ  DXP  RETURN TO CALLER                            CPUMTR    4933
2124  36550                    IX5  X5+X0  COUNT EXCHANGE                           CPUMTR    4934
        0200002123 +           JP  DXJ                                              CPUMTR    4935
                       **      SXJ - SIMULATED EXCHANGE EXIT TO PROGRAM MODE.       CPUMTR    4937
                                                                                    CPUMTR    4938
                                                                                    CPUMTR    4939
2125  5160000075       SXJ     SA6  MR  SET EXCHANGE ADDRESS                        CPUMTR    4940
        0200000076             JP  MR+1  EXIT TO WAIT FOR *MTR*                     CPUMTR    4941
                                                                                    CPUMTR    4942
```

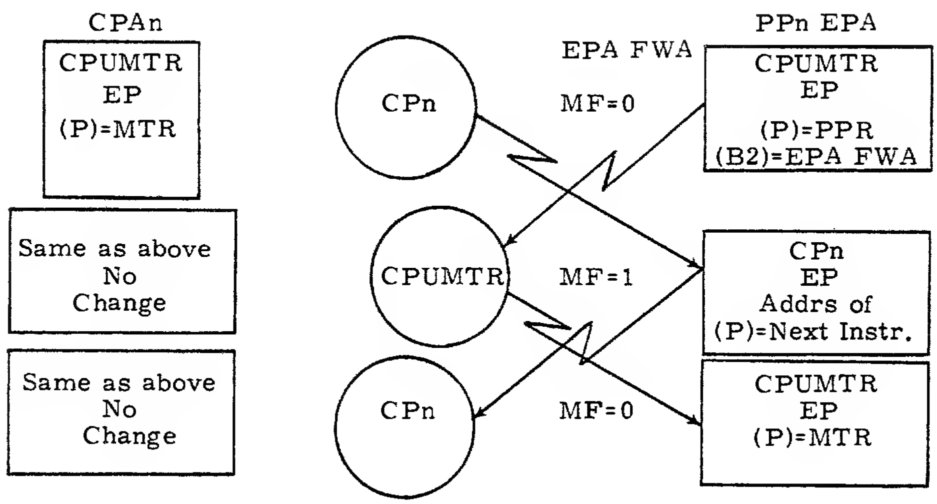
Figure 3-7. Part 1 - Exchange Packages Defined

```
2126           SXJL   BSS  0                                              CPUMTR   4943
               **     PXP - PPU EXCHANGE PACKAGE.                         CPUMTR   4945
               *      COPIED ONCE FOR EACH PPU.                           CPUMTR   4946
               *      (A5) = PPU OUTPUT REGISTER ADDRESS.                 CPUMTR   4947
                                                                          CPUMTR   4948
                                                                          CPUMTR   4949
2126      524         BSS  20*21B  SPACE FOR 20 PPUS IF NEEDED            CPUMTR   4950
                                   (Never more than 19 used since PP0 is not
                                   defined here.)
2652           PXP    EXP  P=PPR,FL=(,MCM),FLX=(,MEC),B1=1, B2=PXP        CPUMTR   4951
               **     SXP - SYSTEM JOB EXCHANGE PACKAGE.                  CPUMTR   4953
               *      COPIED TO SYSTEM CONTROL POINT.                     CPUMTR   4954
                                                                          CPUMTR   4955
                                                                          CPUMTR   4956
2672           SXP    EXP  P=/PROGRAM/PRG,FL=(,MCM),FLX=(,MEC),           CPUMTR   4957
                      MA=(,SCA),B1=1
```

Figure 3-8. Part 2 - Exchange Packages Defined

3.3.1 Pool PPU Request

Assume the CPU is active with CPn and MF=0. If MF=1, then the exchange will not take place. PPn will build a CPUMTR EP in its EPA.



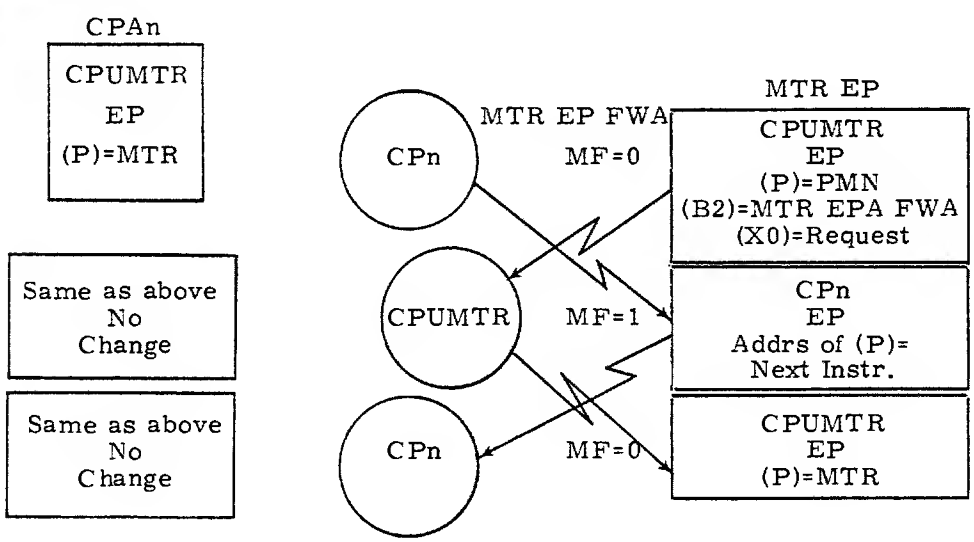
1. PP sets word zero of EP. (P)=PPR, (B0)≠0 (B2)=EP address for the PP issuing MXN.
2. CPUMTR starts executing at PPR. When complete, it issues XJ B2
3. (P)=MTR since this location follows MTRX in CPUMTR. The next time this PP calls CPUMTR, it will reset (P)=PPR.

Figure 3-9. Pool PPU Request

NOTE: If the PP function requires a response in its OR (output register), CPUMTR will exit to MTRP which will fall into MTRX. If no response is required, CPUMTR will exit to MTRX. MTRX is just an XJ B2. MTR follows MTRX; therefore, after the exchange, (P)=MTR in the CPUMTR and EP in the PPn EPA. Refer to Figures 3-4 and 3-40.

3.3.2 PPMTR Request

This is the same as the pool PP request except that (P)=PMN and (X0)=request in the MTR EPA.



1. MTR sets up P, B0, B2. The request is stored in (X0) and the MTR issues MXN.
2. CPUMTR starts executing at PMN and exits at MTRX
3. Same as pool PP.

Figure 3-10. PP MTR

The following figure graphically shows the relationships of the monitors, poll PPs, and CPs.

Figures 3-11, 3-12, 3-13, and 3-14 shows the 4 types of exchanges in detail.

Type of Exchange	WHO does	WHAT to	WHOM,	WHEN,	WHY and	WHERE	for which final DISPOSITION
3	CP Prog	request	CPUMTR	whenever	need help	RA+1	CPUMTR/PP
4	System Prog.	request	CPUMTR	time to quit	to end	PX	CPUMTR
1&2	Pool PPs/MTR	request	CPUMTR	whenever	need help or inter-lock function 35-71.	OR	CPUMTR/PP
	Pool PPs	request	MTR	whenever	need help or inter-lock function 1-34.	OR	MTR
2	MTR	spec. request	CPUMTR	whenever	need help	XO in EP	CPUMTR

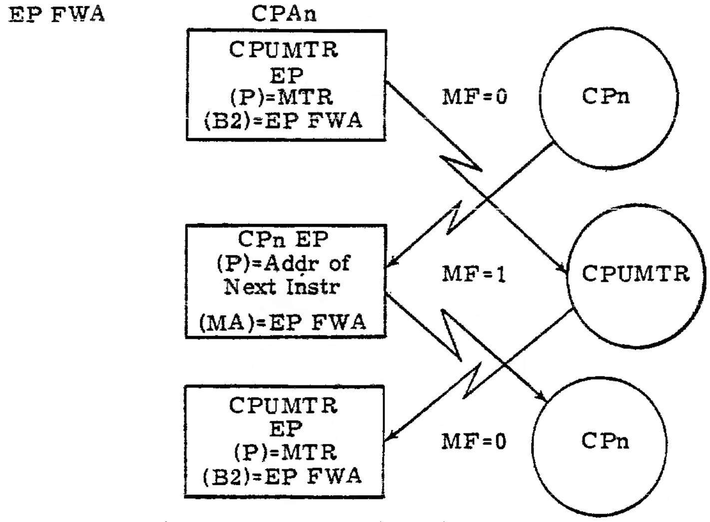
There are only 4 types of exchanges in KRONOS/NOS.

1. Pool PP
2. MTR
3. CP Prog.
4. System CP n+1

Figure 3-10-1. Relationship of the Monitors, Poll PPs and CPs

3.3.3 Program Request

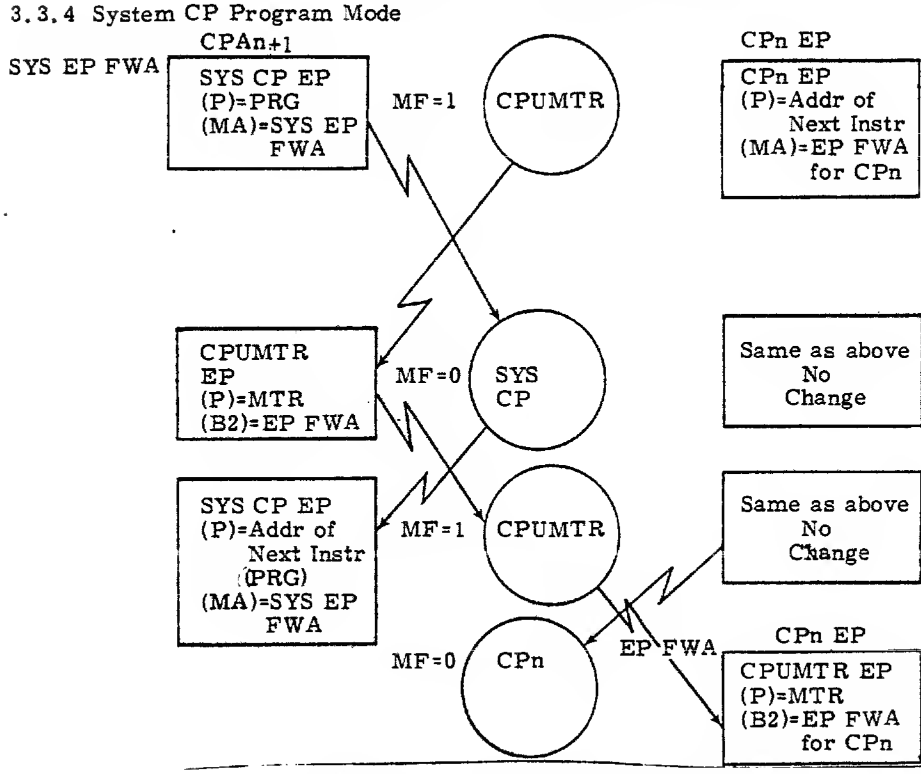
Since CPn is running in CPU (MF=0), MA=the address of CPn and CPA=EP FWA.



1. CPn places the request in RA+1, and will either XJ (MA) (where MA is the hardware register in the CPU), or wait for MTR to notice the request.
2. CPUMTR will process the RA+1 request and (unless recall is requested) will reactivate CPn by XJ B2. (Note: (B2)=EP FWA)
3. CPUMTR will exit at MTRX which will set (P)=MTR in CPn CPA.

Figure 3-11. Program Request

3.3.4 System CP Program Mode



1. CPUMTR will add this request to the system CP queue. It then exits to MTRX (which is an XJ), thereby setting (P)=MTR in the EP.
2. When the system CP has exhausted its queue, it will XJ (MA) back to CPUMTR.
3. System Control Point has finished and exchanges to CPUMTR.
4. CPUMTR will now start the highest priority CPn.

Figure 3-12. System CP Program Mode

NOTE: The SYS CP can be interrupted by a PP program. In this case the PPn EPA will contain the SYS CP EP of which the (P)= address of next instruction to execute (not PRG).

3.4 EXAMPLE OF SYSTEM INTERACTION

A probable sequence of system interaction is illustrated and explained in flow diagrams Figure 3-13 through Figure 3-22.

3.4.1 Assume CPUMTR is running in MM, and it decides to activate CP12 (i.e., give the CPU to CP12).

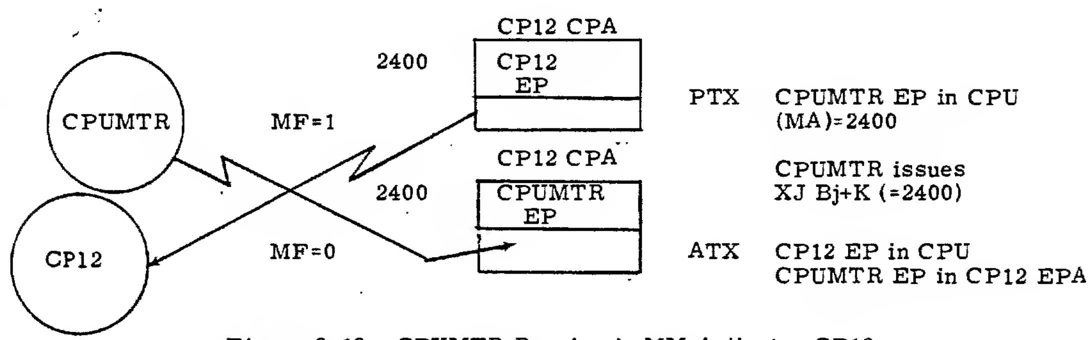


Figure 3-13. CPUMTR Running in MM Activates CP12

3.4.2 Assume PP3 asks CPUMTR to perform a function. PP3 must build a CPUMTR EP in PP3 EPA. Note that RA=0, FL= machine field length, and P=PPR, the FWA of CPUMTR PP function processor. PP3 will issue MXN. Since MF=0, this exchange will occur.

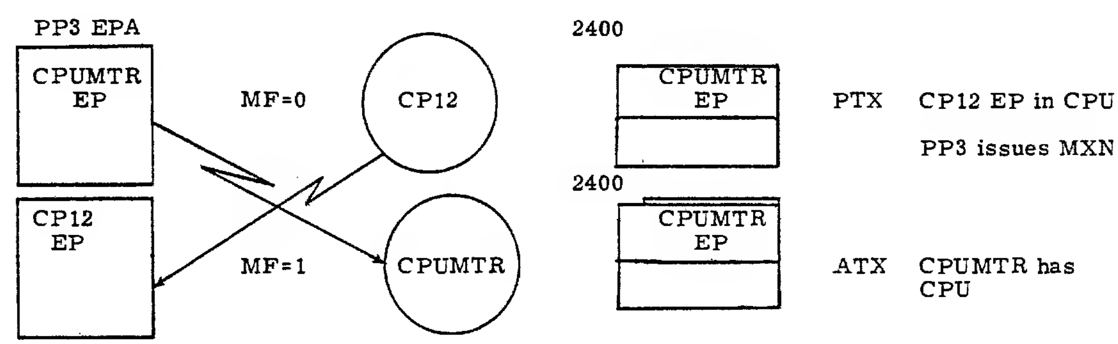
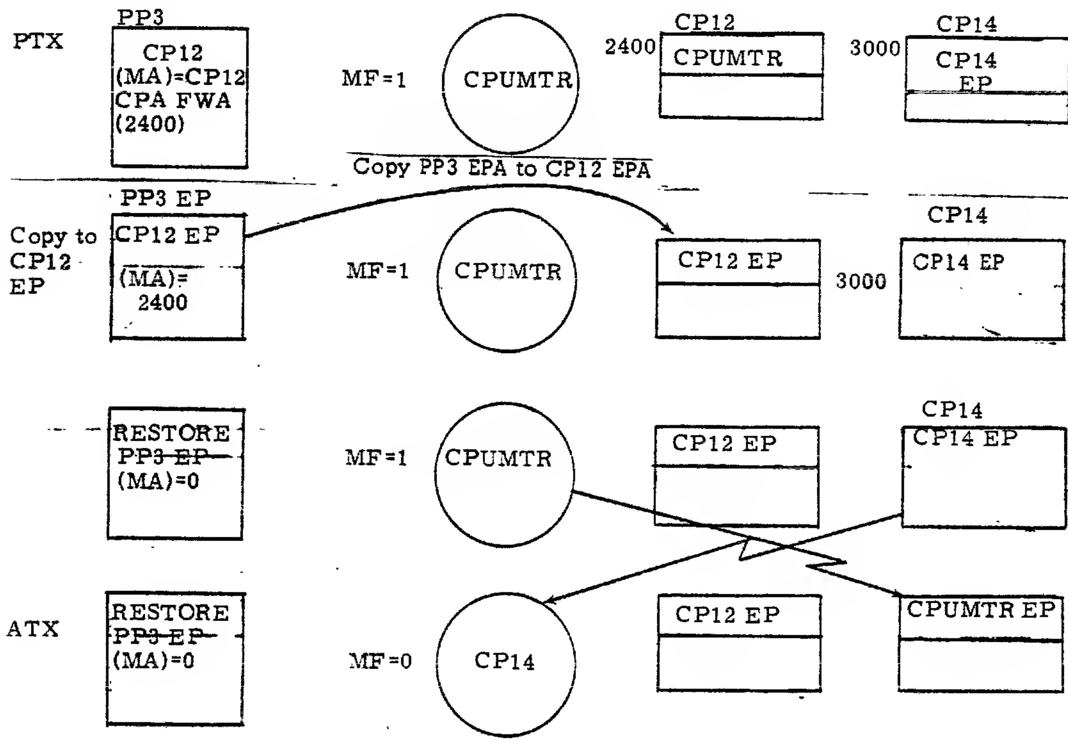


Figure 3-14. PP3 Requesting Function from CPUMTR

3.4.3 Suppose CPUMTR processes the PP request and then determines from CPU priorities that CP 14 should be activated.



NOTE: CP14 area may exist from a previous XJ by MTR or may have been built due to a request by the scheduler or the advancement routines. Since CP12 will not be activated, it is necessary for CPUMTR to move CP12 EP from PP3 EPA to CP12 EPA before issuing XJ Bj+K (=3000).

Figure 3-15. CPUMTR Processing PP Request Activates CP14

3.4.4 Suppose MTR decides to switch CPs (i.e., stop CP14 and start CP10) and issues an ACSF (switch job request) to the CPUMTR. MTR must build a CPUMTR EP in his EPA and issued MXN.

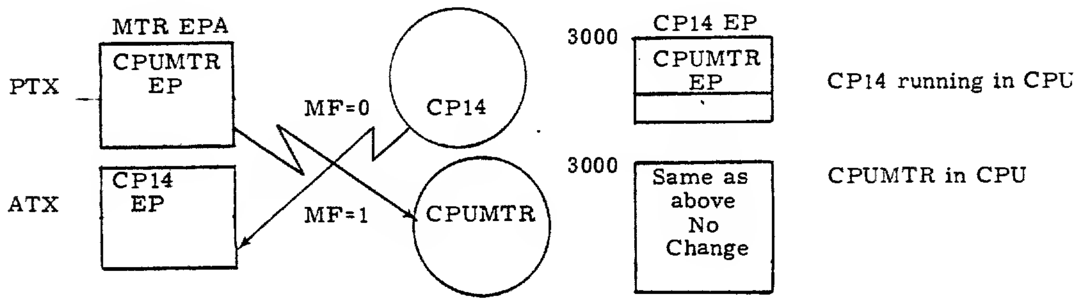


Figure 3-16. MTR Switches CPs

3.4.5 Then CPUMTR will activate CP10. MTR decides which CP to start, and CPUMTR will start it.

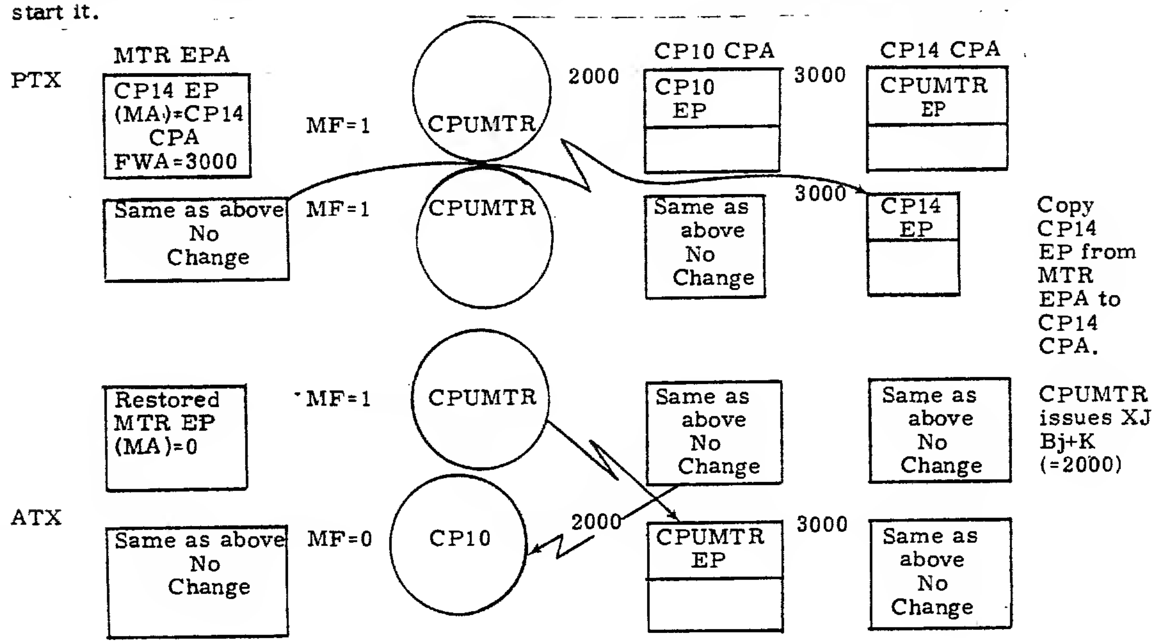
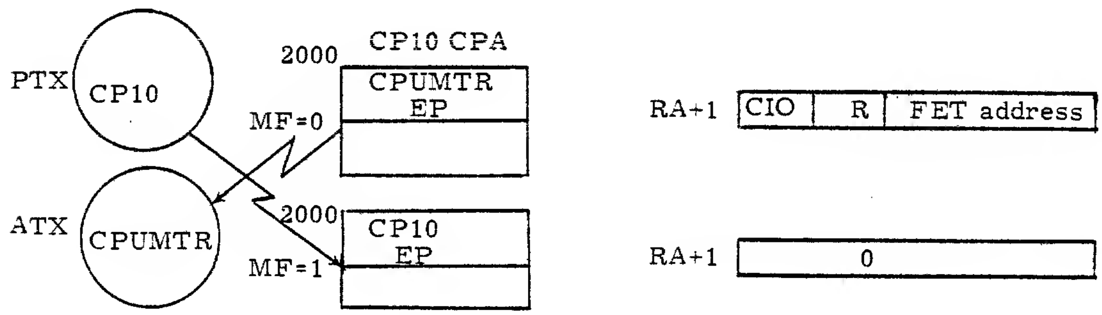


Figure 3-17. CPUMTR Activates CP10

3.4.6 Suppose CP10 wishes to call CIO. CP10 places the call in RA+1 and issues XJ.

Since MF=0, the exchange will store the CPU EP value in location (MA). Now, whenever CPUMTR built CP10 EP, he set (MA)=2000 and (P)=MTR, the FWA of CPUMTR CP request processor.



NOTE: Now, CPUMTR places CP10 into auto recall, calls CIO to a pool processor, say PP6, and searches for the highest CPU priority job to activate which is CP16.

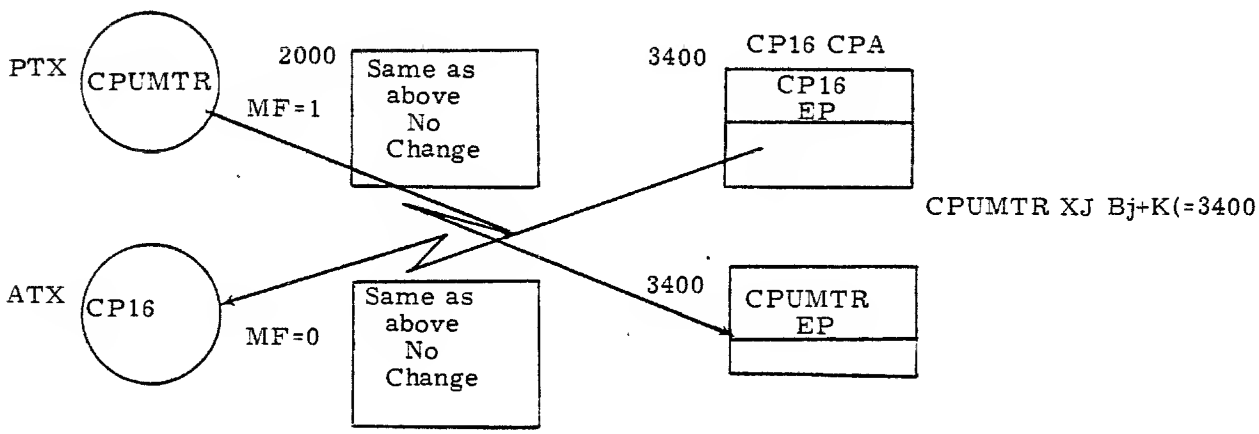


Figure 3-18. CP10 calls CIO and CPUMTR, Places CP10 into Autorecall, Calls CIO and Activates CP16.

3.4.7 Suppose CIO runs to completion, sets the status of its operation to complete, and prepares to drop. In order to drop, CIO will MXN to monitor with a DPPM (drop PP request).

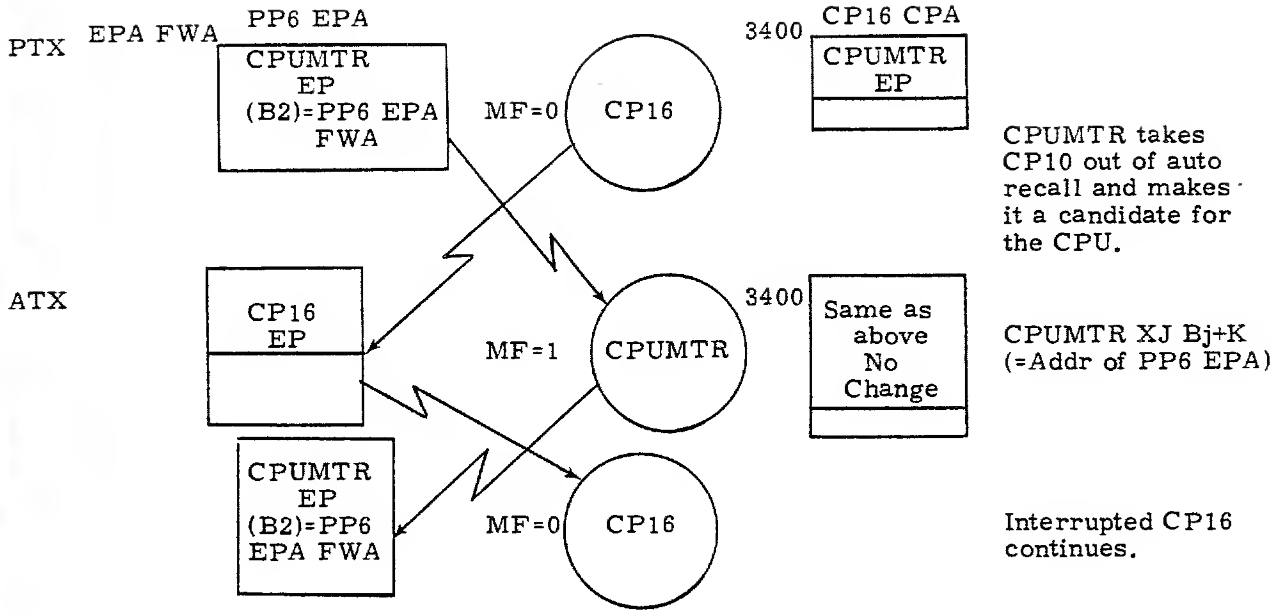


Figure 3-19. CI0 Runs to Completion and MXNs to Monitor

3.4.8 Suppose PP4 issues a DTKM (drop track function) via an MXN.

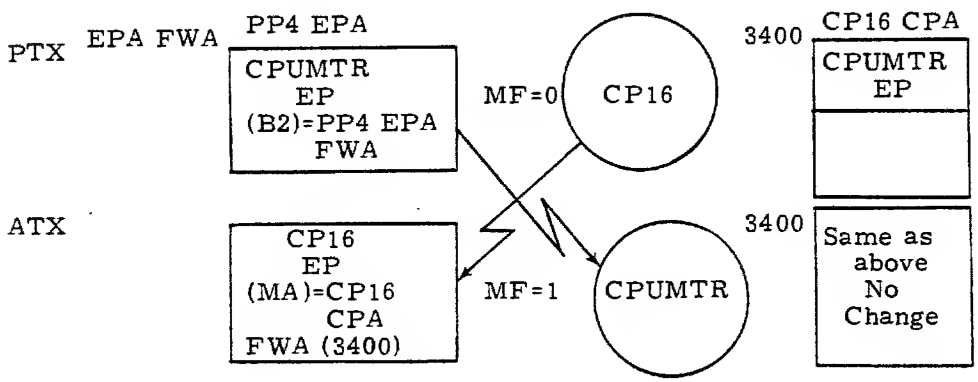


Figure 3-20. PP4 Issues DTKM Via MXN

3.4.9 Now PP4 will idle on its OR until monitor satisfies its request. DTKM is a request which takes too long a CPU time-slice, therefore, it is processed by CPUMTR in program mode via the system CP. The system CP is treated as any other CP except that it has the highest priority. CPUMTR will begin processing this request by queuing the request and XJ Bj+k (=4000), thereby activating the system CP. If the system CP is interrupted, CPUMTR will process the interrupting request.

If it is a request which is also processed by the system CP, CPUMTR will queue this request and reactivate the system CP. In this way, all these types of requests are handled in a first come, first served order.

Before the exchange can occur, however, CPUMTR must copy CP16 EP from PP4 EPA.

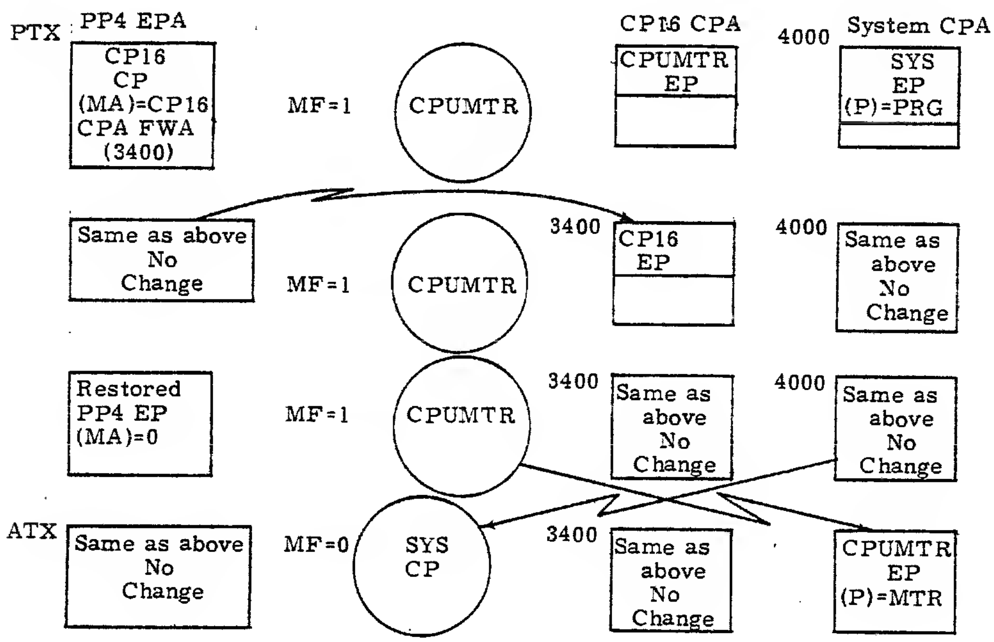
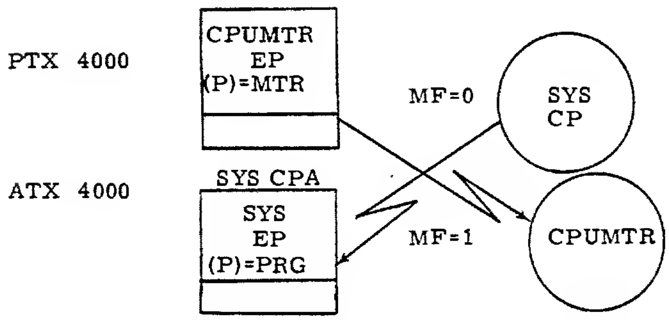


Figure 3-21. System Control Point Processing

3.4.10 When system CP completes all the requests in his queue, he will XJ (MA) to the CPUMTR.

NOTE: For system CP (MA)= 4000, and CPUMTR will have set (P)=MTR in the CPUMTR EP at system CPA. When the system CP exchanges, CPUMTR will begin at MTR. However, the system CP begins executing at PRG.



Now, CPUMTR will pick a CP to activate, and on and on.

Figure 3-22. System Control Point XJ (MA) to CPUMTR

3.5 SUB CONTROL POINTS (SCP)

The SCP concept can best be thought of as a mini operating system operating with a particular CP. The CP user may define a controlling segment – similar to an absolute overlay level (0, 0) – as the CP Executive (E). The E can be thought of as a monitor for this CP. Subroutines (similar to absolute overlays) may be defined as subtasks and can be run with complete protection from each other.

The E may time share its CP time slice among a series of subtasks. Since the SCP has its own RA and FL, the E can have a hardware protected CM area. The E, since its RA and FL encompass all of the SCP's FL, may directly oversee and control all the SCPs defined at its CP. The E must load the subtask into an SCP. Since the subtask is a (0.0) overlay, LDR cannot be used (see section 12.1). Instead, the E can load the subtask with a READSKP or READ CIO request.

Whenever the E decides to start up an SCP, it sets up the Exchange Package (EP) for this SCP, sets RA+1 to XJP and exchanges to CPUMTR. CPUMTR will use the SCP EP and exchange in the SCP after validating the EP set MA=SCP. Now the SCP will run with its own RA and FL in the hardware CPU registers. (RA and FL must be within the CPs RA and FL or CPUMTR will abort the CP.)

The SCP is an absolute overlay which must be loaded by the E prior to starting the SCP up. When the SCP is done, it may set RA+1 and/or exchange back to CPUMTR. CPUMTR will

exchange in the E, which then may process the request or exchange to CPUMTR with this request in E's RA+1. E must pluck the request from the RA+1 relative to the SCP.

The SCP will exchange to the E under any of the following conditions.

1) SCP has exceeded its CPU time slice.
2) SCP enters a request into its RA+1 and/or executes XJ.
3) CPU detects an error (arith, out of bounds, etc.) and sets the error flag.

In summary, the SCP concept allows:

1) An Executive (E) program to be protected from subprograms
2) The E has complete control of subprogram EP
3) Subprograms are essentially relocatable overlays
4) The system will let E process SCP RA+1 requests
5) Any number of sub control points may be used
6) E has time limit control
7) E is restarted if SCP issues a XJ or RA+1 request
8) The SCP status is shown on the console

The format of the XJP and SCP EP is shown below.

XJP RA+1 request

	18	6	18	18
(RA+1)=	XJP	0	time	addr

XJP = in display code

time = CPU time limit for SCP in milliseconds. i.e., interrupt this SCP and exchange to E after the SCP has used time milliseconds

addr = address relative to this CP of the SCP EP

Response

(X2)=	milliseconds of CPU time before this call to the SCP
(X6)=	2000B+EF* RA of SCP
(X7)=	milliseconds of CPU time used by SCP

*EF= error flag caused by SCP, the use of 2000B+EF allows the E to do an UX_i to get the EF value into a B register and have the RA in an X register.

NOTE: Total CPU time used by this CP at this point is X2+X7.

	59	53 47	35	17 0
n		P	A0	B0
n+1		RA_{CM*}	A1	B1
n+2		FL_{CM*}	A2	B2
n+3	EM	EM	A3	B3
n+4	RA_{ECS*}		A4	B4
n+5	FL_{ECS*}		A5	B5
n+6	MA**		A6	B6
			A7	B7
	X0			
	X1			
	X2			
	X3			
	X4			
	X5			
	X6			
n+15	X7			

*These values must be within the bounds of the CP at which the E is executing.
**This field is controlled by CPUMTR. In this way the RA+1 requests of the SCP will be sent to the E.

Figure 3-23. Sub-Control Point Exchange Package (SCP EP)

Sub-control points, as the name implies, are divisions of a Central Memory Control Point. A programmer can set up a control point to contain 2 or more programs; one of these will be designated as the "executive" which will monitor the other program(s) which are known as sub-control points.

The executive controls its sub-control points in much the same manner that the system monitor controls the control points. When a control point makes a system request or exceeds its time limit or makes an error, control is given back to the system monitor. Similarly, when a sub-control point makes a system request or exceeds its time limit or makes a CPU error, control is given back to the executive. The executive sets up each sub-control point so that, within the field length of the control point, each sub-control point has its own "RA" and "FL" and cannot go outside its boundaries. The executive is thus protected from access by the sub-control points, whereas the executive's RA and FL define the full control point so the executive can watch over and control all sub-control points within the field length.

3.5.1 Implementation

The sub-control point concept depends on the executive program's handling of the sub-control points. This involves starting, stopping, error processing and other functions similar to those of the system monitor.

Just as the system monitor keeps track of each control point through its exchange package, the executive can control the sub-control points through their exchange packages.

It is the responsibility of the executive to set up an exchange package for each sub-control point; each exchange package must have the appropriate RA, FL, P, etc., for the sub-control point. These exchange packages must be set up somewhere within the executive's field length, but probably not within the field length of the sub-control point. To start execution of a sub-control point, the executive uses an XJP request indicating the address of the exchange package area of the sub-control point to be activated. When CPUMTR picks up the request, it terminates the executive and activates the sub-control point described in the exchange package area indicated on the XJP request. CPUMTR also sets a flag in the Control Point Area showing that at this control point a sub-control point is now active. Once activated a sub-control point runs until:

1. it makes a CPU error
2. it exceeds its time limit
3. it makes an RA + 1 request

Under any of these three conditions, control is given back to the executive.

The executive can thus monitor error processing for the sub-control points. Errors can be noted and examined without termination of the control point. Upon returning control to the executive, certain information is set up in the X registers:

(X2) = msec before this sub-control point began

(X6) = EF (12 bits) | RA of this sub-control point

(X7) = msec used by this sub-control point

One of the parameters on the XJP request is the time limit for the sub-control point. When this time limit is passed, control goes back to the executive.

When a sub-control point makes an RA+1 request, control is returned to the executive; the executive can then decide whether to:

1. ignore the request
2. handle the request itself
3. pass the request on to CPUMTR using RA+1 of the control point (Executive)

Sub-control points can be set up by any CP programmer using any programming language; some features are only usable by COMPASS programs. The structure of the executive is flexible within the limits we have discussed so far. Two programs using sub-control points in different ways have proved quite useful and are described here to give you some ideas on the design and use of sub-control points.

3.5.2 Examples

3.5.2.1 TRANEX Overview

TRANEX is designed to let many different users use one system; each user needs transaction processing. Users can set up their own programs for transaction processing and all transactions can be handled through the TRANEX executive.

TRANEX uses sub-control points so that the transaction executive can maintain complete control over each task to be performed. Within TRANEX's field length we need a protected area for the executive and the remaining field length can be used by up to 31 sub-control points. The tasks to be performed require different programs that do not need to be in core simultaneously; rather

than using traditional overlays which have no protected area for the executive, each task or transaction program can be set up as a sub-control point which can be activated as necessary by the executive.

User transaction programs can be written in any programming language. In order to make the programs more useful, the first 100 words of each program should be allocated for communication between sub-control points; this can be done by using labeled common which is always at the beginning of the field length, e.g.,

```
(FTN)    COMMON /CCOMMON/ A(100)
(COMPAS)      USE    /CCOMMON/
              BSS     100
(COBOL)   COMMON  STORAGE  SECTION.
          77  A  OCCURS  100  TIMES.
```

NOTE:

RA+0 through RA+100 is normally not easily available to higher level languages, therefore the technique of labeled common allows, an easy method of access to RA+101 through RA+201.

The user programs should be compiled and then LINK can be used to create a (0.0) overlay from each transaction program.

Each transaction to be processed must give enough information to indicate the proper transaction program to be brought in for processing. This information could include:

1. user's name (code)
2. type of transaction
3. data to be used in the transaction

The executive will then bring in the appropriate transaction program into TRANEX's field length and set up the program as a sub-control point. Since the user program is an absolute (0.0) overlay the loader cannot be used to load it*, so the executive will have to use a CIO function to bring in the program. The executive will also have to set up an exchange package for the sub-control point and put any necessary information into the 100 word communication area in the sub-control point's field length. If the transaction requires another program to complete

*LDR will always give control directly to the (0.0) overlay after loading; this will not allow the Executive to start the sub-control point.

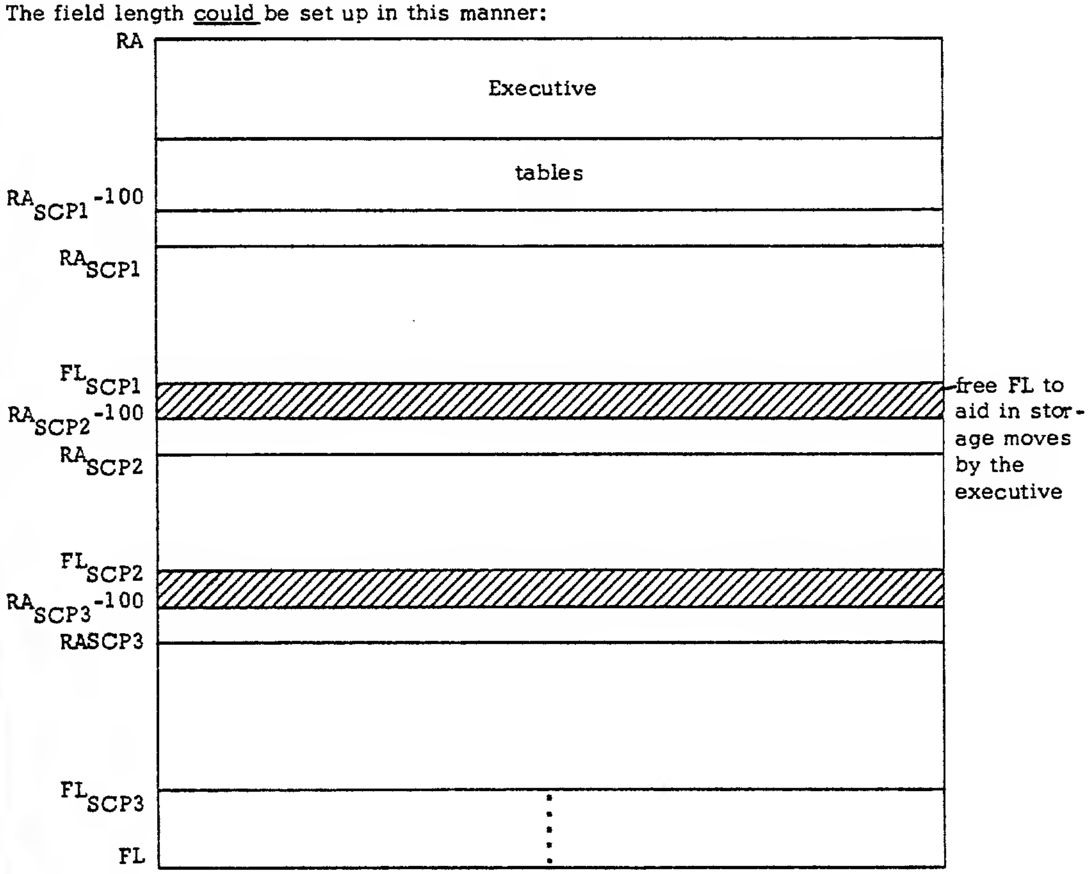
the task, a request must be make to the executive to bring in the other program. The executive always checks to see of the program is available in core already and brings in a copy if necessary; then the executive copies the appropriate data from the communications block of the calling sub-control point to the communications block of the called sub-control point.

3.5.2.2 TRANEX executive

The TRANEX executive's job is to set up the field length in the most efficient way. The field length must contain:

- the executive's code
- tables
- sub-control points
- exchange package areas for each sub-control point

The field length could be set up in this manner:



The area RA_{scp}-100 through RA_{scp} can be used for the exchange package area for the sub-control point. The executive can fill in this area as it reads in the program; it gets P from the 50 table of the (0,0) overlay binary, it can set up values for the registers for COMPASS programs, it sets up RA and FL depending on where the program was read into memory and how many words were read in.

The executive always checks through its tables to see if the program is already at a sub-control point; if it is already at a sub-control point, the executive checks to see if it is a re-usable program; if the program is not in memory or not re-usable, the executive will read in another copy of it. The executive looks for the next available place in memory to put the program and brings it in using READR (READSKP) and updates its tables. The executive must set up the exchange package and can then start execution of the sub-control point by making an XJP request with the address of the exchange package area. When CPUMTR picks up the request it exchanges in the sub-control point and sets the flag in the Control Point Area to indicate that there is a sub-control point active at the control point.

3.5.2.3 TRANEX Sub-Control Points

TRANEX sub-control points are all (0,0) absolute overlays. These programs are loaded by the executive using a CIO' function. The executive also sets up an exchange package for each sub-control point so that each sub-control point can use only memory within its own RA through RA+FL-1.

TRANEX has set up one sub-control point (ITASK) which decides which other program needs to be brought in to handle a transaction. ITASK can look at the transaction code from the user and find the name of the program to do the task. Since ITASK is a sub-control point itself and cannot go outside its own field length, ITASK must ask the executive to activate the appropriate transaction program at a sub-control point.

When a sub-control point needs assistance from the executive, it puts a request in <u>its own RA+1</u>; this causes an exchange back to the executive. The executive looks at the request and can:

1. ignore the request
2. process the request itself
3. pass the request on to CPUMTR

After the request has been handled, the executive can give control back to the sub-control point if it is appropriate.

An example of a request would be a sub-control point requiring the loading of another sub-control point to complete a task. When the first sub-control point puts the request in its RA+1, the executive is exchanged in; the executive brings in a copy of the program if necessary and copies the communications block from the calling program to the called program. The RA+1 of the sub-control point is within the FL of the executive who can read the request.

Sub-control points can be designed in different ways. TRANEX uses the executive to bring in (0,0) overlays as sub-control points. TUBE uses the loader to bring in both the executive and the sub-control point.

3.5.2.4 TUBE Overview

TUBE is designed so that the sub-control point can be any user relocatable program that has an RJ =XTUBE (CALL TUBE in FTN) instruction in it. The external reference to TUBE will cause the loader to load in TUBE after the user's relocatable program. Within TUBE's code there is an XJP request; the XJP request will exchange in the user program and set the sub-control point bit in the Control Point Area. TUBE will then be the executive and the user program will be the sub-control point; all error processing and system requests will have to go through TUBE. Thus TUBE is set up to monitor and help debug a user program.

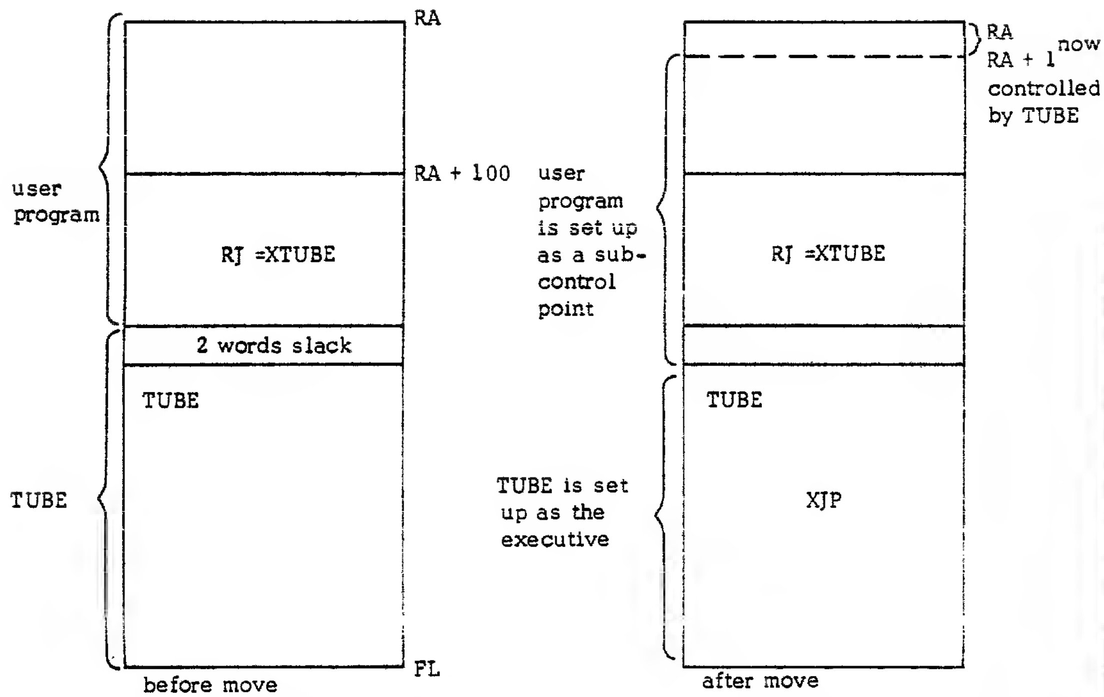
TUBE was designed to simulate the debugging features of DIS, a program available on the operator's console. DIS allows the user to interact with his program using instructions such as:

BKP,a
ENX_i
ENP
GO (start the SCP)
etc.

3.5.2.5 TUBE Implementation

TUBE is brought in by the loader when a user program includes an RJ =XTUBE. TUBE will be loaded after the last word of the user program.

The first two words of TUBE's field length are slack words; when TUBE is entered it moves the user program down 2 words into these slack words. This means that RA and RA+1 are no longer available to the user program and TUBE will have to set up and process all RA+1 requests for the user program which is now set up as a sub-control point.



TUBE then sets up an exchange package area within its field length for the sub-control point. In this exchange package RA will be set to 2 because TUBE moved the program down 2 words; P can be set to the address stored at address TUBE from the RJ (+2).

TUBE then can issue an RA+1 request with XJP and the address of the exchange package area it set up for the sub-control point. Now TUBE is designated the executive and the user's program is the sub-control point. The sub-control point bit will be set in the Control Point Area and all error processing and system interface will go through TUBE.

In order to give control back to the user's program as a regular control point program rather than as a sub-control point, TUBE will have to move the program back up and fix up the exchange package area so that FL is the original FL. Then TUBE issues an XJR request specifying the address of the exchange package area for the user's program. When the XJR request is picked up the exchange packages will be switched (i.e., from the EPA in the FL to the EPA in CPA) which will activate the user's program, but the sub-control point bit will not be flipped on.

3.5.2.6 TUBE's Structure

TUBE is set up, as previously mentioned, with 2 words of slack. These 2 words are used in moving the user program down away from RA and RA+1 so that TUBE can have control of these locations.

TUBE also has a buffer area which it uses as the exchange package area for the sub-control point. TUBE must set up an exchange package for the user program setting RA=2 etc., to reflect the move.

Besides these buffer areas TUBE also includes code that simulates DIS features. TUBE has routines for interpretting TTY input as DIS interprets console input. For example, from the console the operator can enter values for the A and X and B registers using the instructions ENX_i, ENA_i, ENB_i. With TUBE, the user can enter values from the TTY using ENX_i or ENA_i or ENB_i; TUBE reads the value from the TTY, writes it into the exchange package area for the sub-control point and then issues an XJP request which exchanges in the sub-control point with the new values for the registers (when ready) to start the program in "DIS" mode).

When the user's program has been initialized with an XJP it will run as a sub-control point until:

1. it exceeds its time limit
2. makes a CPU error
3. makes an RA+1 request

Upon any of these conditions, control is given back to TUBE, the executive. TUBE also includes code to handle these conditions.

When a sub-control point makes a request at its RA+1, TUBE is exchanged back in and it can either:

1. handle the request itself
2. pass the request on in its own RA
3. ignore the request

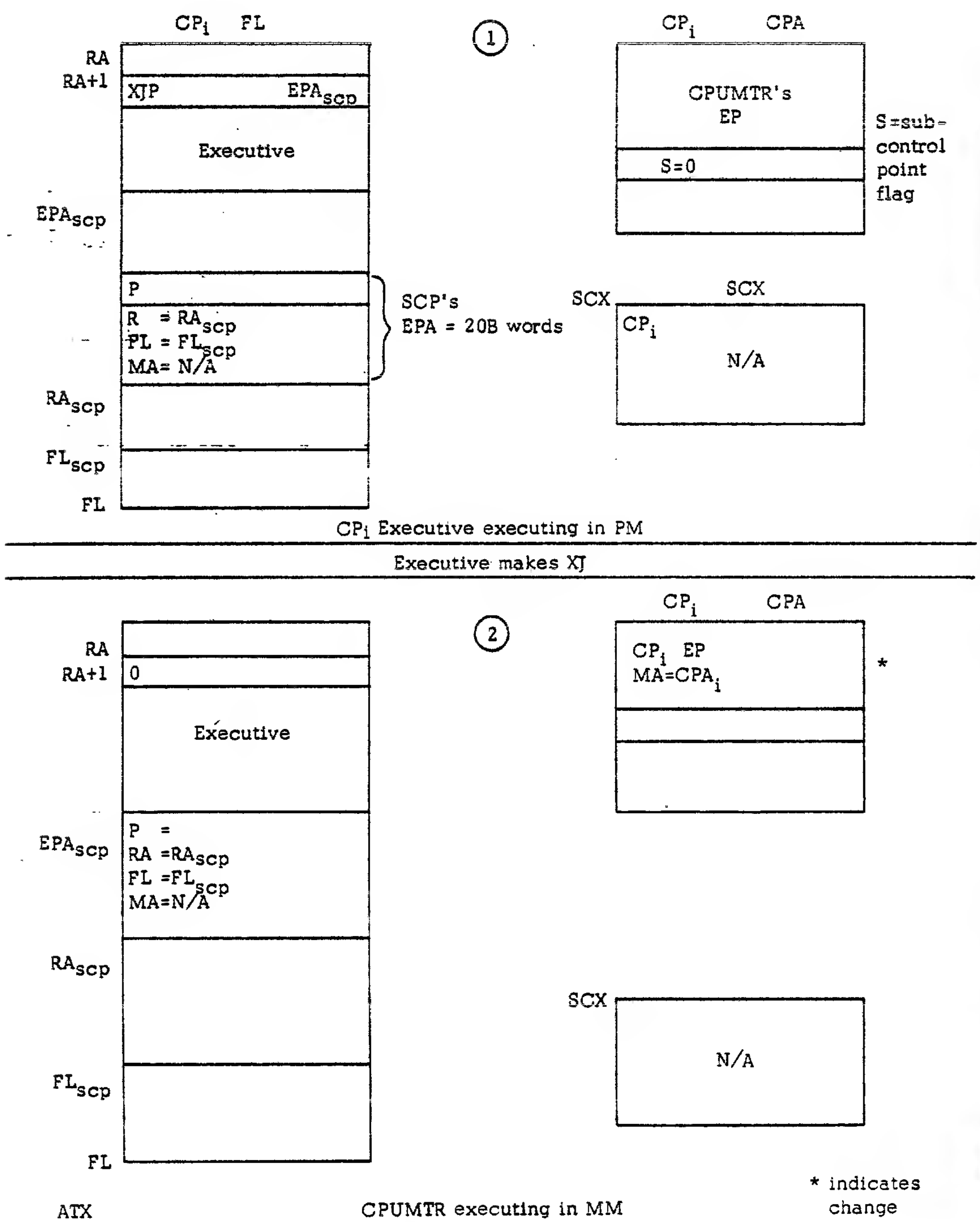
Thus TUBE monitors all requests and can catch errors if they occur in this area.

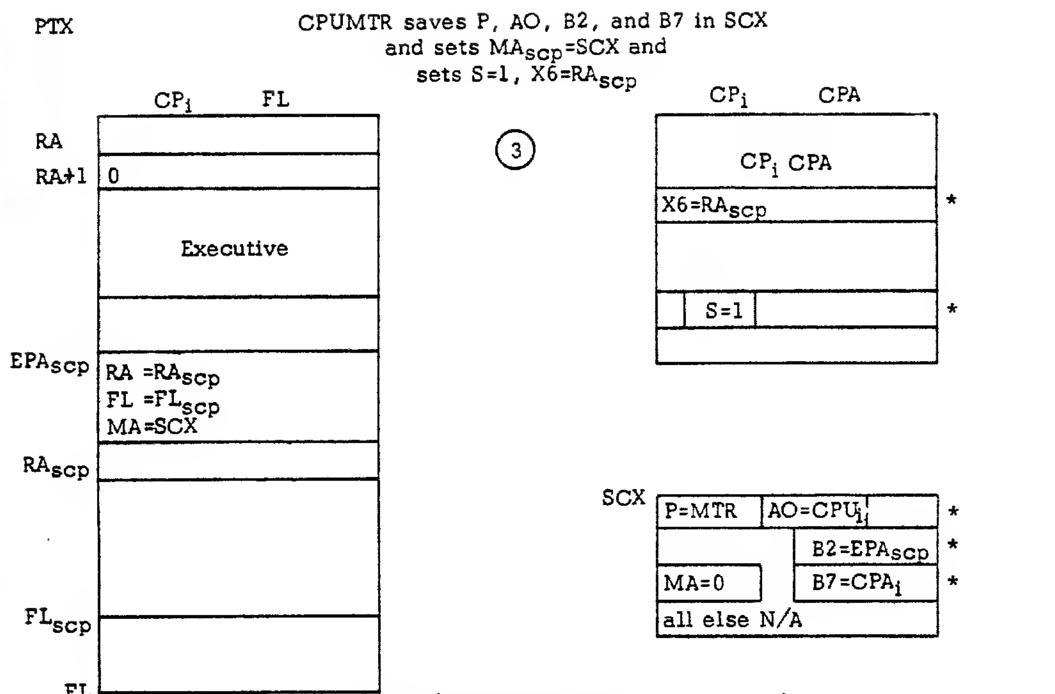
TUBE also monitors CPU errors. It prints out a message on the TTY and will accept input from the user to determine what step should be taken next.

Since TUBE is the executive it can access the entire field length and can modify code within the user program. In this way the BKP feature is easily implemented by saving the BKP instruction word and setting it to zero. When the BKP occurs the word can be restored. Hence, TUBE can BKP on an instruction address, next or last address or on RA+1 requests.

TUBE uses the capabilities of the executive to monitor and modify a user progra. This shows how the sub-control point concept is a convenient tool for testing and debugging programs.

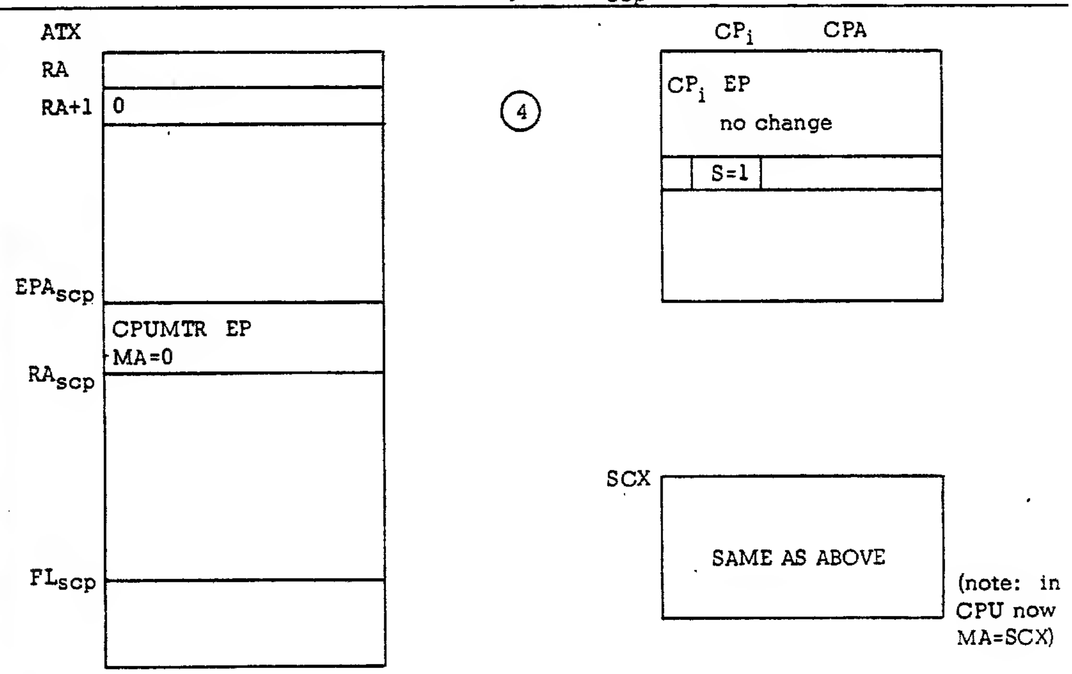
3.5.3 Sequence of XJP for Sub-Control Points Showing What SCX in CMR is used for PTX.



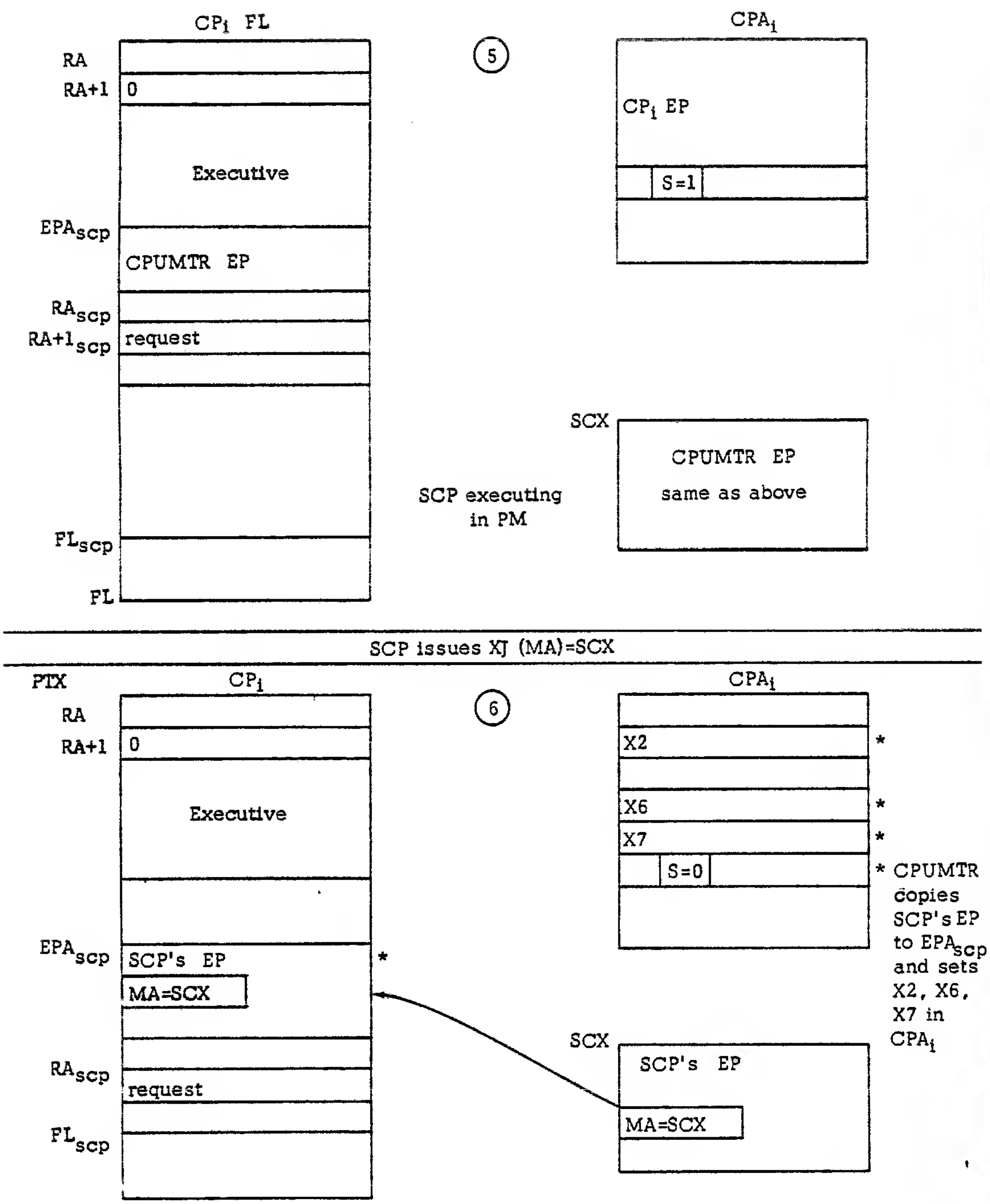


CPUMTR still in MM (setting up SCX and CPA)

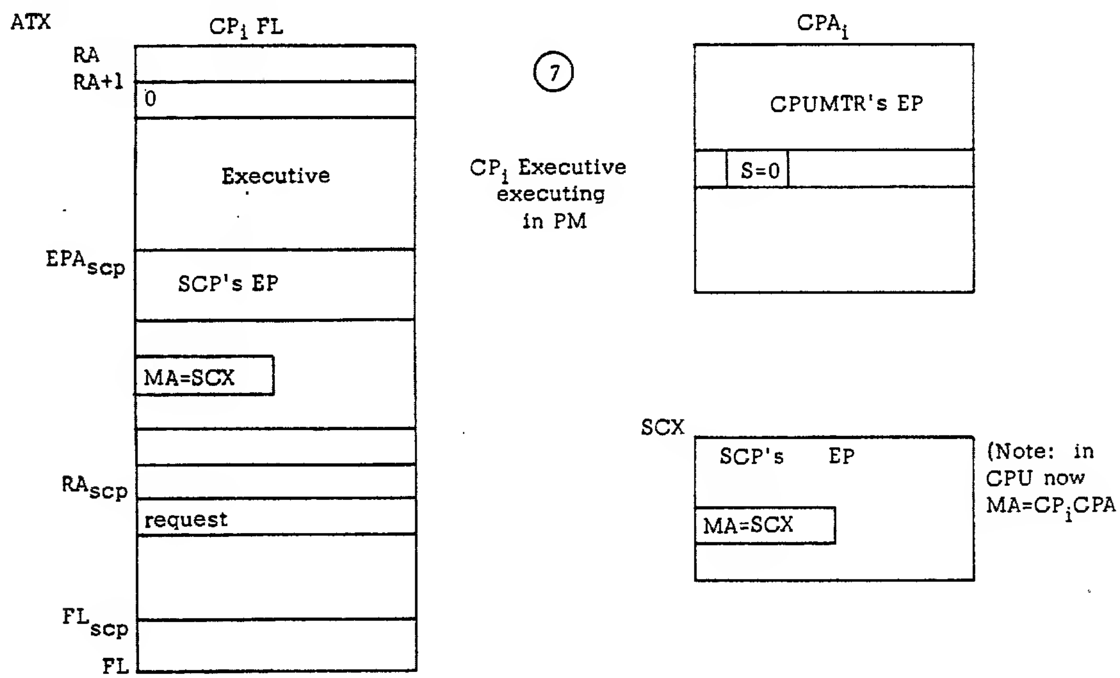
CPUMTR XJ B2=EPAscp



Now SCP executing in PM


CP$_i$ FL
RA
RA+1
0
Executive
EPA$_{scp}$
CPUMTR EP
RA$_{scp}$
RA+1$_{scp}$
request
FL$_{scp}$
FL
5
CPA$_i$
CP$_i$ EP
S=1
SCX
CPUMTR EP
same as above
SCP executing in PM
SCP issues XJ (MA)=SCX
PTX
CP$_i$
RA
RA+1
0
Executive
EPA$_{scp}$
SCP's EP
MA=SCX
*
RA$_{scp}$
request
FL$_{scp}$
6
CPA$_i$
X2
X6
X7
S=0
*
* CPUMTR copies SCP's EP to EPA$_{scp}$ and sets X2, X6, X7 in CPA$_i$
SCX
SCP's EP
MA=SCX

CPUMTR XJ B2=CP_iCPA



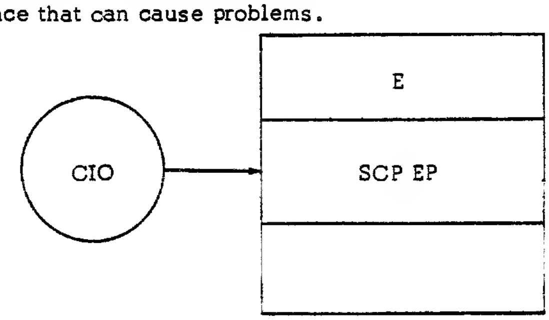
3.5.4 Reasons for Using SCX

1. P=MTR, AO=CPU number and B7=CPA B2=EPA_{scp} are the only parts of the SCP EP that are important to insure that it can not be changed.

2. IF SCP EP is only saved in Es FL then some PP routine working for the E while SCP executes could clobber SCP EP in Es FL but would never disturb SCX. Hence, we protect the system from a possible XJ which moves garbage into the CPU registers. (Hence protecting CPUMTR exchange package.)

3. CPUMTR must copy SCX back into SCP EP in ES FL so that the E can read it and/or write to it and restart the SCP without completely rebuilding it.

4. Possible sequence that can cause problems.



   a. E calls CIO without Auto-recall.

   b. E calls XJP to start SCP.

   c. E has inadvertently specified a buffer which includes the SCP EP and it is clobbered by CIO.

   d. SCP makes an XJ and if MA=SCP EP in Es FL, CPU hardware registers get garbage and system is destined to crash, since CPUMTR exchange package living in SCP EPA. However, since MA=SCX in CMR, this problem is circumvented and no system crash is forecast.

5. The point of setting S=1 (sub-control point activity flag) allows.

   a. MTR to get SCPs RA from X6 in CPA.

   b. CPUMTR to know that an SCP versus a CP is making a request.

3.6 MTR – PP MONITOR

MTR is loaded into PP0 at dead start time and remains there for the duration of system execution.

MTR performs the following functions:

1. Process certain PPU requests
2. Allocation of central memory
3. Check the CPU for arithmetic errors CD=0
4. Maintain the real-time clock
5. Check (RA+1) of active central programs for system requests
6. Check the status of active control points, so that he can call 1AJ if zero status or rollout status on a CP
7. Checks OR of each pool PP.
8. Start ISJ periodically.

3.6.1 Starting MTR at Dead Start Time

MTR is loaded in PP0. The first location of the code is:

```
T0   CON   PRS-1.
```

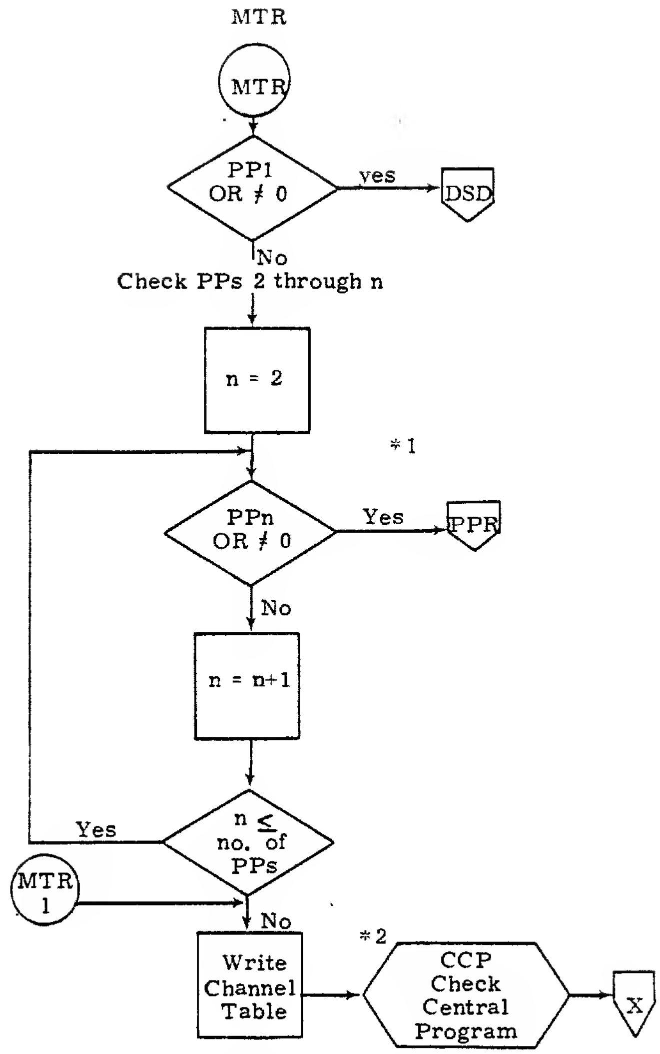
This forces the constant PRS-1 to fall into T0. At the end of the load (P) will be set to (T0)+1 which will be (P)=PRS, the MTR preset routine. PRS will preset all tables and constants. It will create the following tables at the end of the channel table (TCHS) which is generated in the code.

TPPR	Table to hang illegal requests
PPR	Table of request processor address

PRS will overlay itself with the following tables

TMSD	Table of mass storage devices
TMSP	Table of mass storage space available
TSYS	Table of system devices
TFUN	Table of release functions
TUFL	Table of unassigned field length

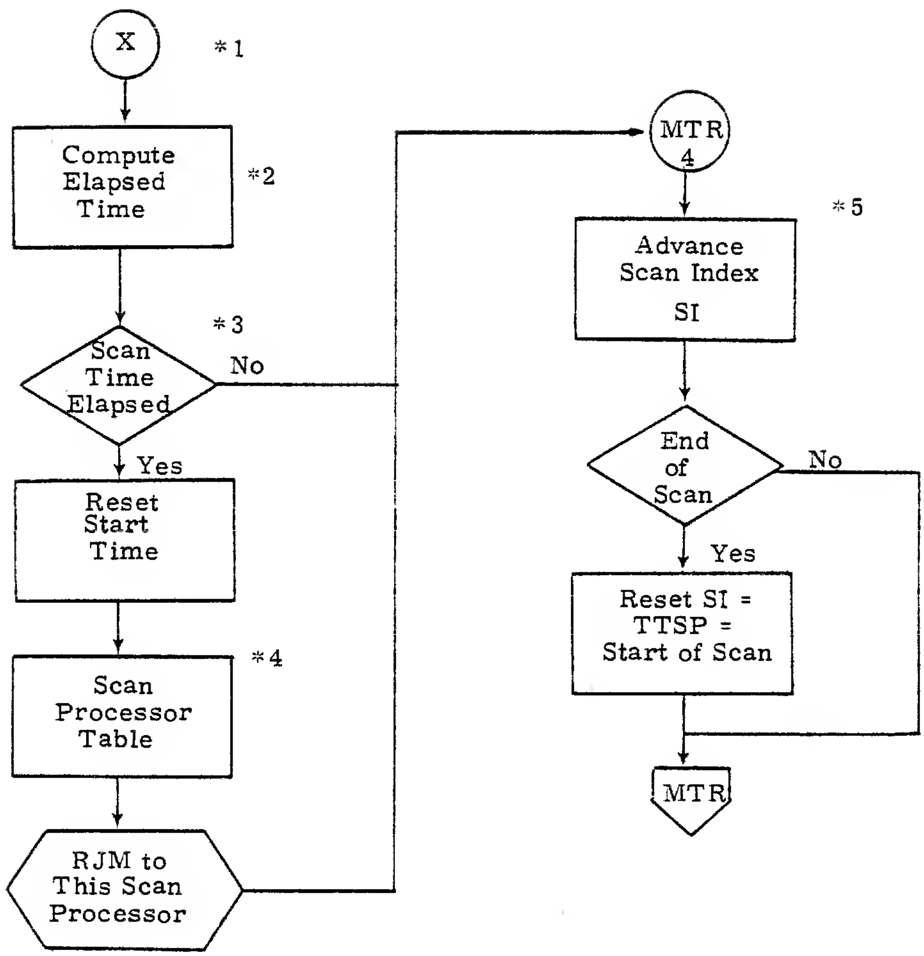
and a message buffer in the top end of core. Then it will effect a jump to INT which will initialize all the tables. When INT is complete, it will jump to the main loop MTR.



*1 This simulated loop is actually a DUP statement in MTR code.

*2 When MTR releases a channel, he sets a flag. At this time, the reservation byte in the channel table in CMR is actually cleared.

Figure 3-24. Main Loop for PP MTR



*1 These operations are performed for all scanners and at any one time, this operation is for the current scanner.

*2 Current time – start time

*3 Elapsed time – delay time

*4 See *1 scan processor Table 3-6

*5 Scan index is saved in cell S1.

Figure 3-25. Process Time Dependent Scanners

TABLE 3-6. TIMED SCAN PROCESSOR *1

Symbol	Value		Description
TTSP	FWA of ART	*2	Advance running times (ART)
	0		Start time
	1000D		Delay time in milliseconds
	0		
.JAC	FWA of JAC		Check job activity (CPU time slice)
	0		
	0		
	0		
.JAS	FWA of JSW		CPU switching (CPU slot time)
	0		
	0		
	0		
.CRC	FWA of CRC		CPU recall (periodic recall)
	0		
	0		
	0		
.PPL	FWA of PPL	*3	PP recall (process PP recall register) and AUTO-RECALL for CPU
	0		
	0		
TTSPL	0		End of table

*1 Each time through the MTRs major loop, only one time dependent scanner will execute. In addition, the fortunate scanner will scan only a specified number of CPs. In Figure 3-25 (*4), MTR will specify the CP to start scanning, and how many CPs to scan. On return from the scanner, MTR will save the number of the last CP scanned in order to restart next time through the loop.

*2 ART copies the delay time from CMR word MSCL into the 4 other processors table. These delay times can be changed by the operator and ART will change them for the processors.

*3 PPL will start up any PP whose name occurs in the PP recall register word RLPW in control point area.

3.6.2 Real-Time Clock

The real-time clock starts with power on and runs continuously. It may be read by any peripheral processor with an input to A (70) instruction from Channel 14B. This channel is separate from the data channels.

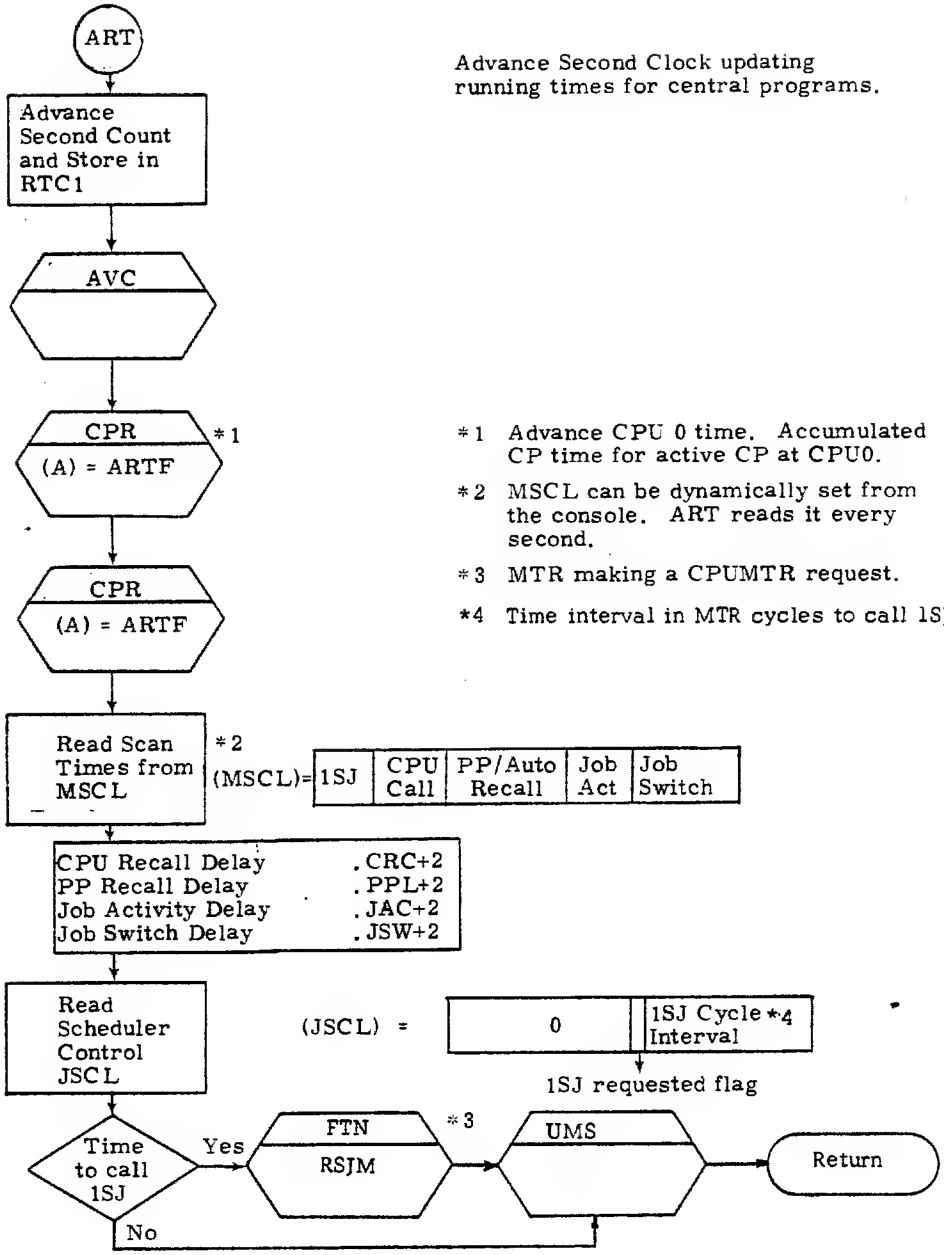
The clock period is 4096D (10,000B) major cycles*. It is a 12-bit register that is advanced each microsecond from 0000B through 7777B. When it reaches 7777B, it starts over at 0000B. It must, therefore, be read at least every 4.096D milliseconds for accurate timing.

The AVC (Advance Clock subroutine) updates the clock. AVC must be entered at least once every 4 milliseconds. In case AVC is called too often, AVC will check the constant MLSC, and, if the elapsed time from last call is less than MLSC, it exits without updating the clock. AVC will update its own internal clock whenever called, unless called before MLSC has elapsed. If RTC has advanced at least 1 millisecond, AVC will update the real-time clock in CMR, RTL word 106, Figure 2-2.

---

* 1 Major Cycle = 1 Microsecond = 1000 Nanoseconds.
1 Minor Cycle = 0.1 Major Cycle = 100 Nanoseconds.

Advance Second Clock updating running times for central programs.



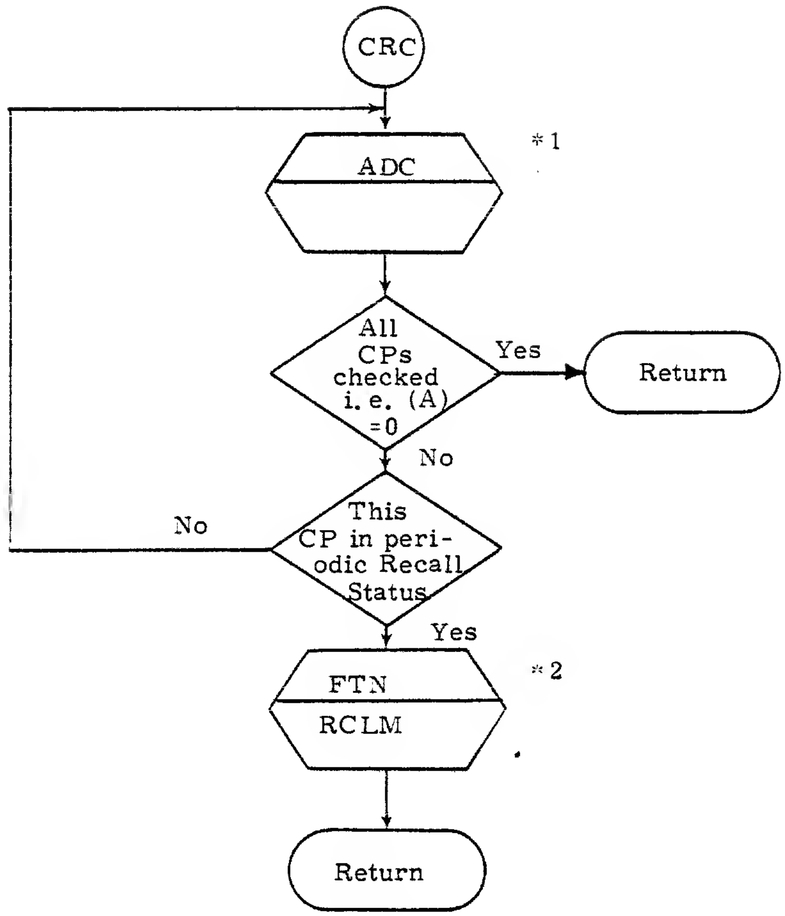
*1 Advance CPU 0 time. Accumulated CP time for active CP at CPU0.

*2 MSCL can be dynamically set from the console. ART reads it every second.

*3 MTR making a CPUMTR request.

*4 Time interval in MTR cycles to call 1SJ

Figure 3-26. ART Advance Running Times

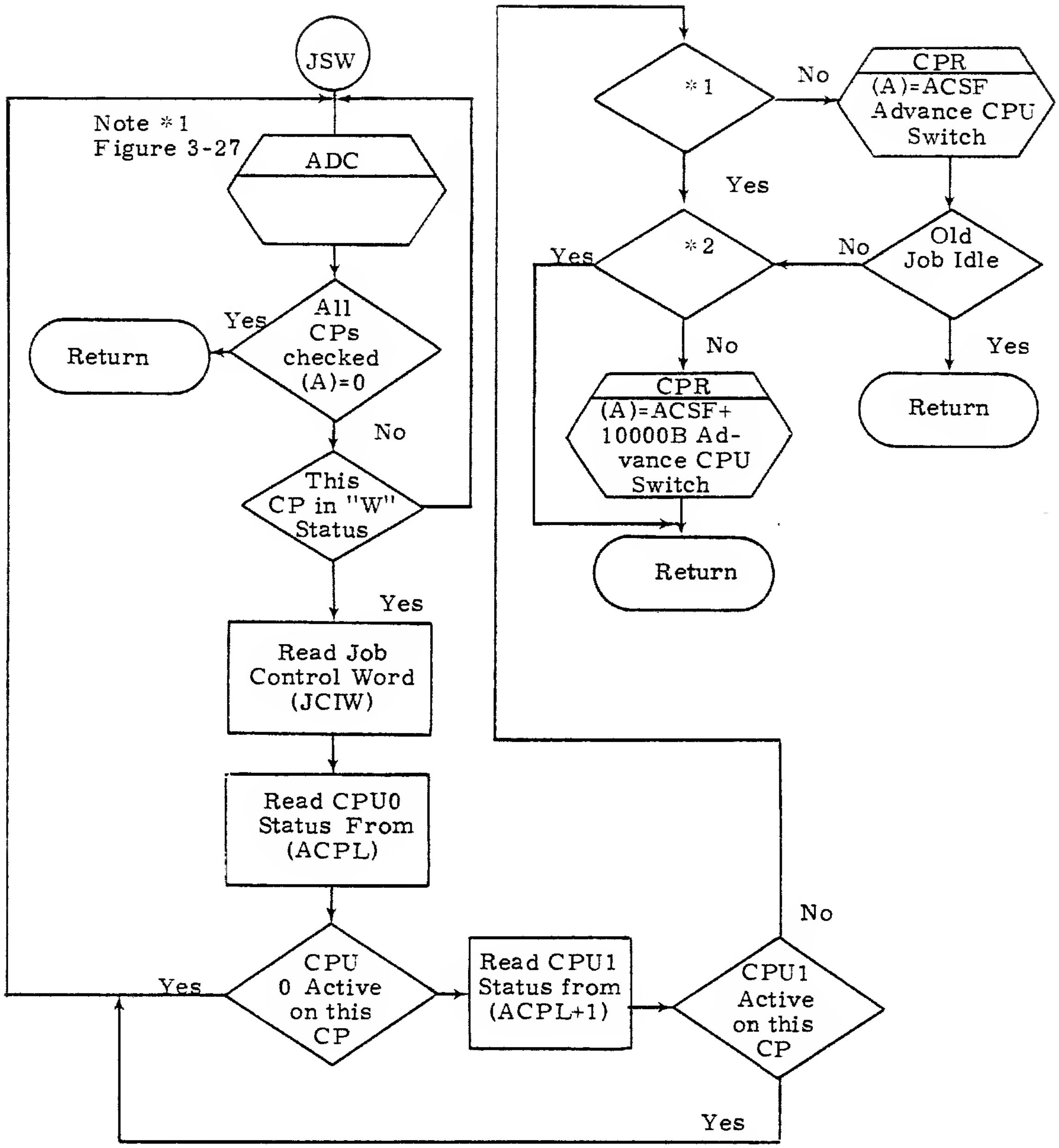


*1 ADC will advance to next CP on exit: ADC considers system control as end of scan and will set (A) = 0.

(A) = 0 if all CPs checked
$\neq$ 0 if active CP found
(CP) = CP number
(CS-CS+4) = CP status

*2 Recall CPU.

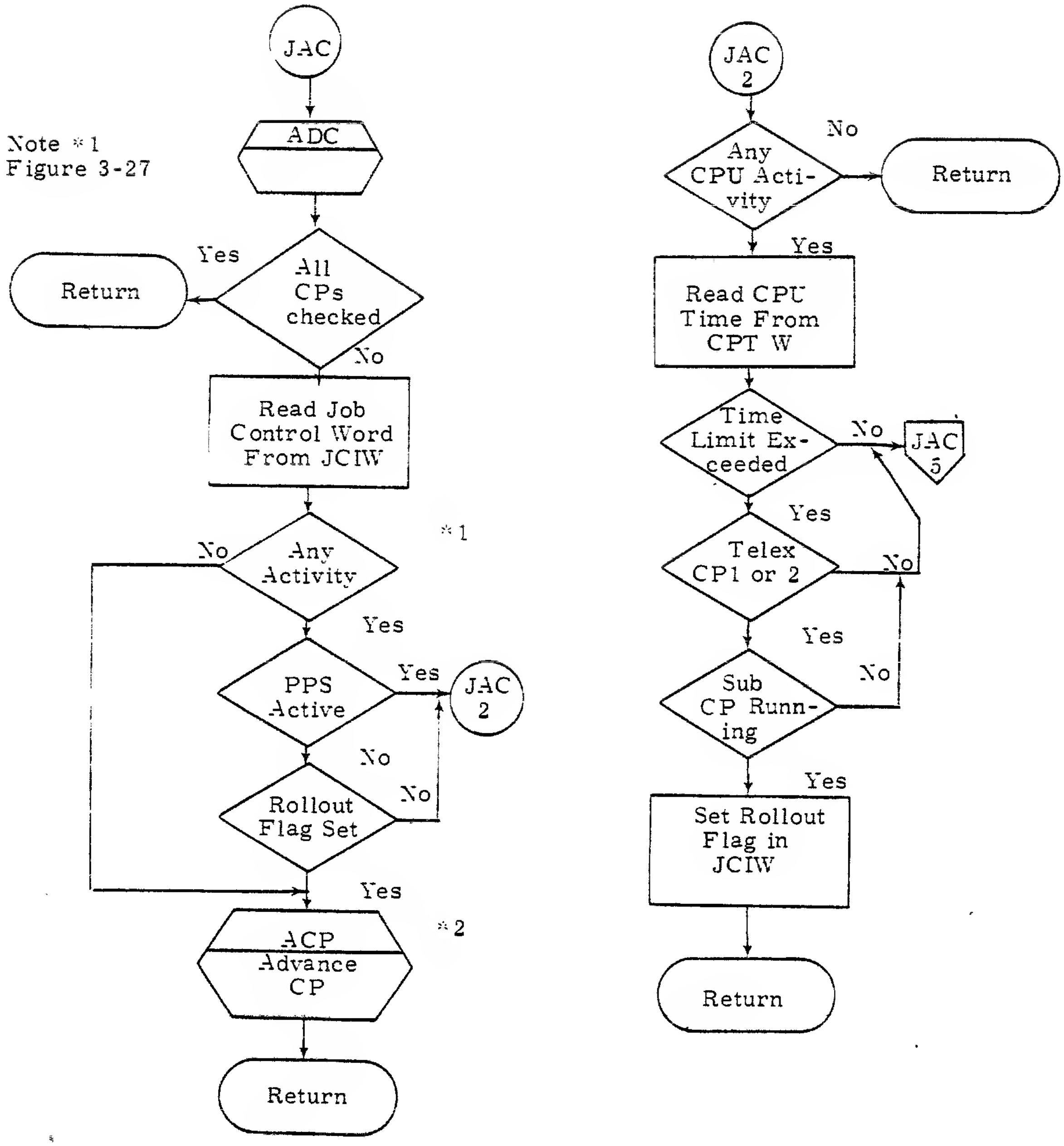
Figure 3-27. (CRC) – Check CPU Recall Status (Periodic Recall)



*1 CPU0 active job CPU priority > this CP CPU priority.

*2 CPU1 active job CPU priority > this CP CPU priority.

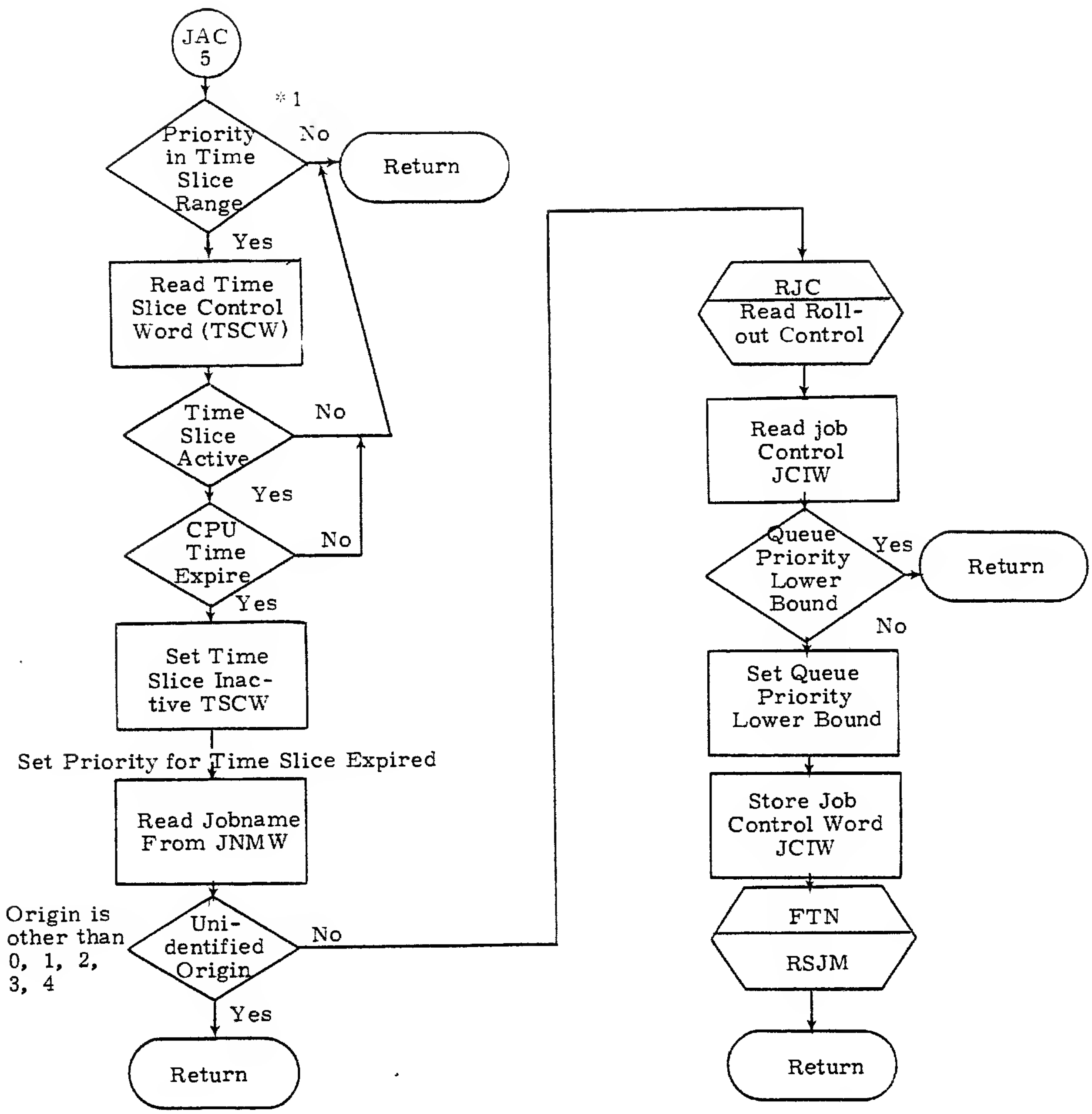
Figure 3-28. JSW – Process CPU Job Switching (CPU Slot Time)



*1 Byte 0 of STSW

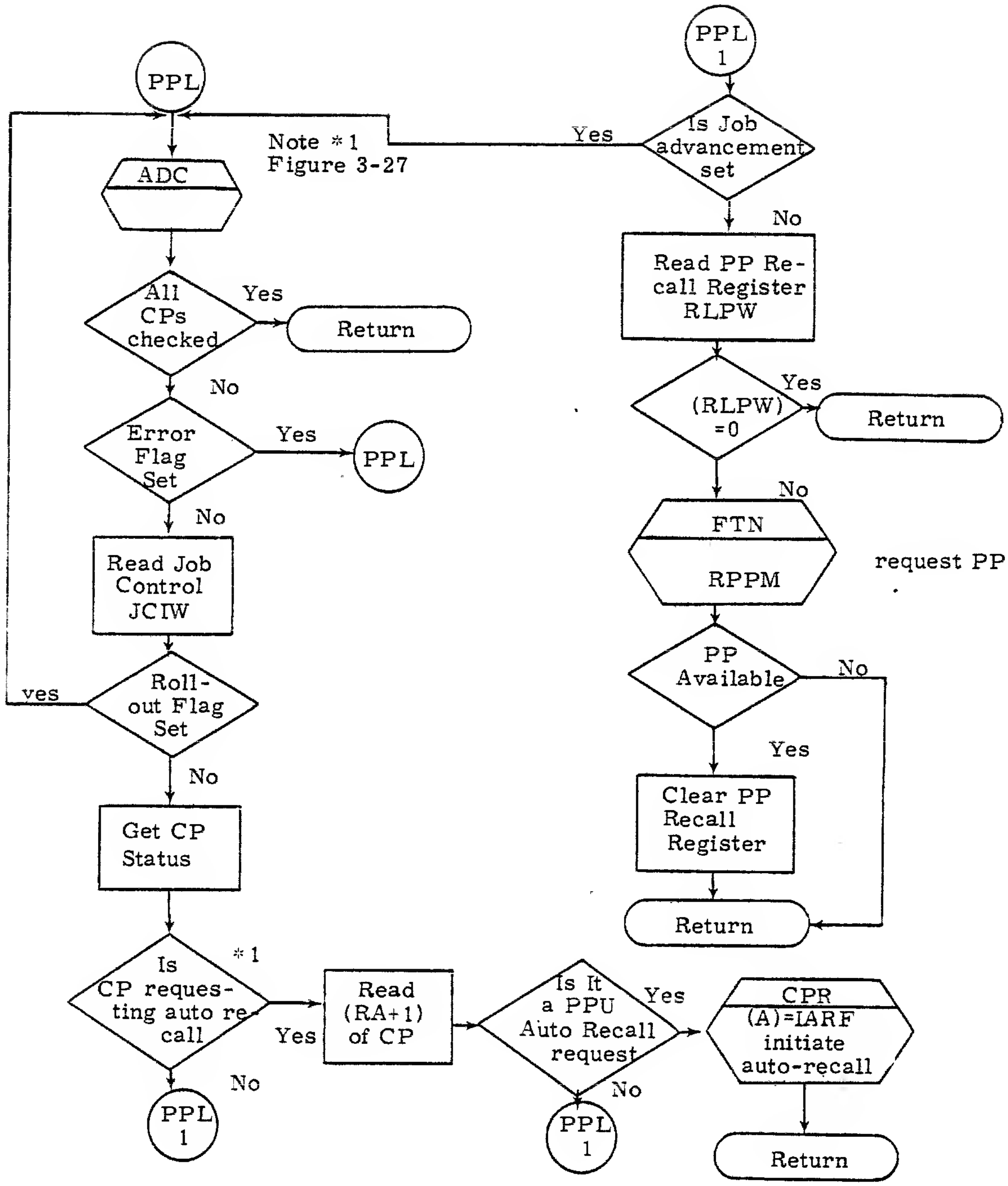
*2 Call up job advance PP routine

Figure 3-29. JAC – Check Job Activity



*1 is MNPS (minimum priority for scheduling) < priority > MXPS (Maximum priority for rollout).

Figure 3-30. Check CPU Time Slice

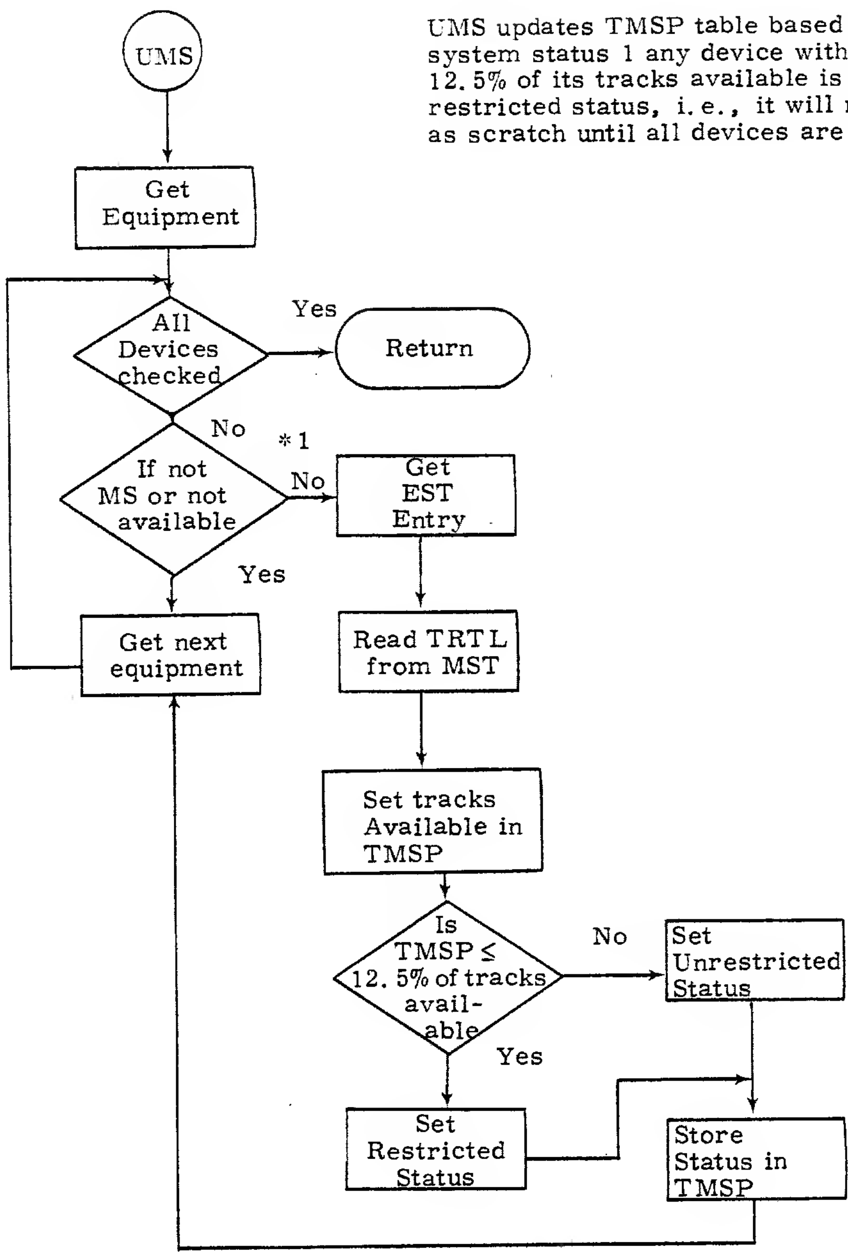


*1 Is CP status bit for auto-recall set.

Figure 3-31. PPL – Process PP Recalls

MTR has many other non-time-dependent routines. A few of them are flowcharted on the following pages.

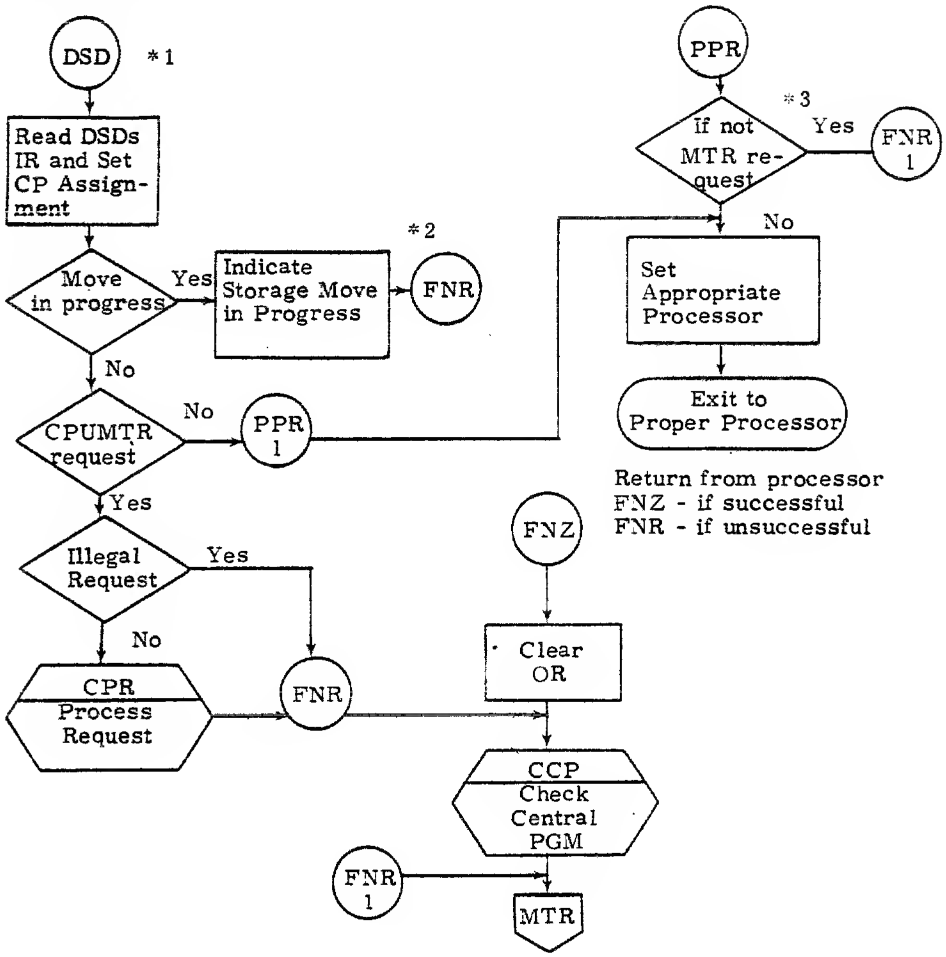
UMS updates TMSP table based on current system status 1 any device with less than 12.5% of its tracks available is put on restricted status, i.e., it will not be used as scratch until all devices are restricted.



*1 MS is mass storage.

Figure 3-32. UMS – Update Mass Storage Tables

PP function requests are made to MTR by placing the function code in byte 0 of the PPs OR. When the request is complete, MTR clears byte 0 of the OR.



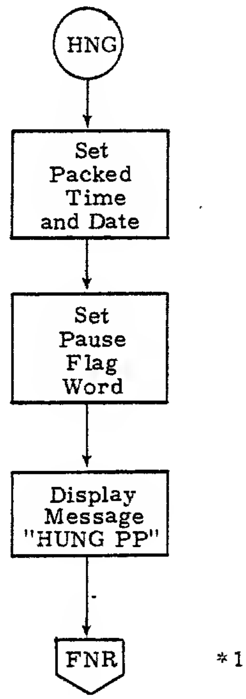
*1 When DSD wants to do an action for a CP (such as n.XXX), he temporarily attaches him-self to that CP by placing the CP number in his IR, then he makes the request.

*2 If this CP is moving the status must be set.

*3 I.E., if request illegal then effectively hang PP since OR is never cleared this will not display PP hung at system PP.

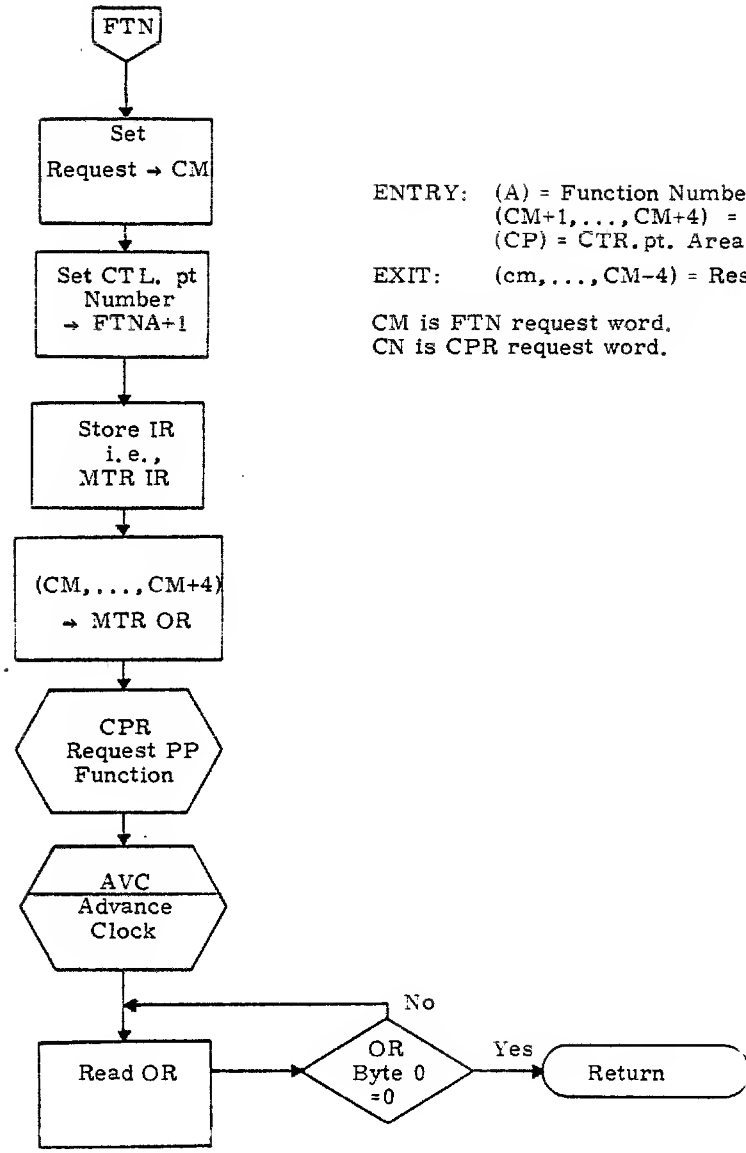
Figure 3-33. DSD PP Function Request

If any of the functions requested desire an illegal operation (for example, DCHM drop channel wishes to drop a channel which does not exist) then it will jump to this routine.



*1 Don't clear OR and thereby hang this PP.

Figure 3-34. HNG – Hang PPU and Display Message



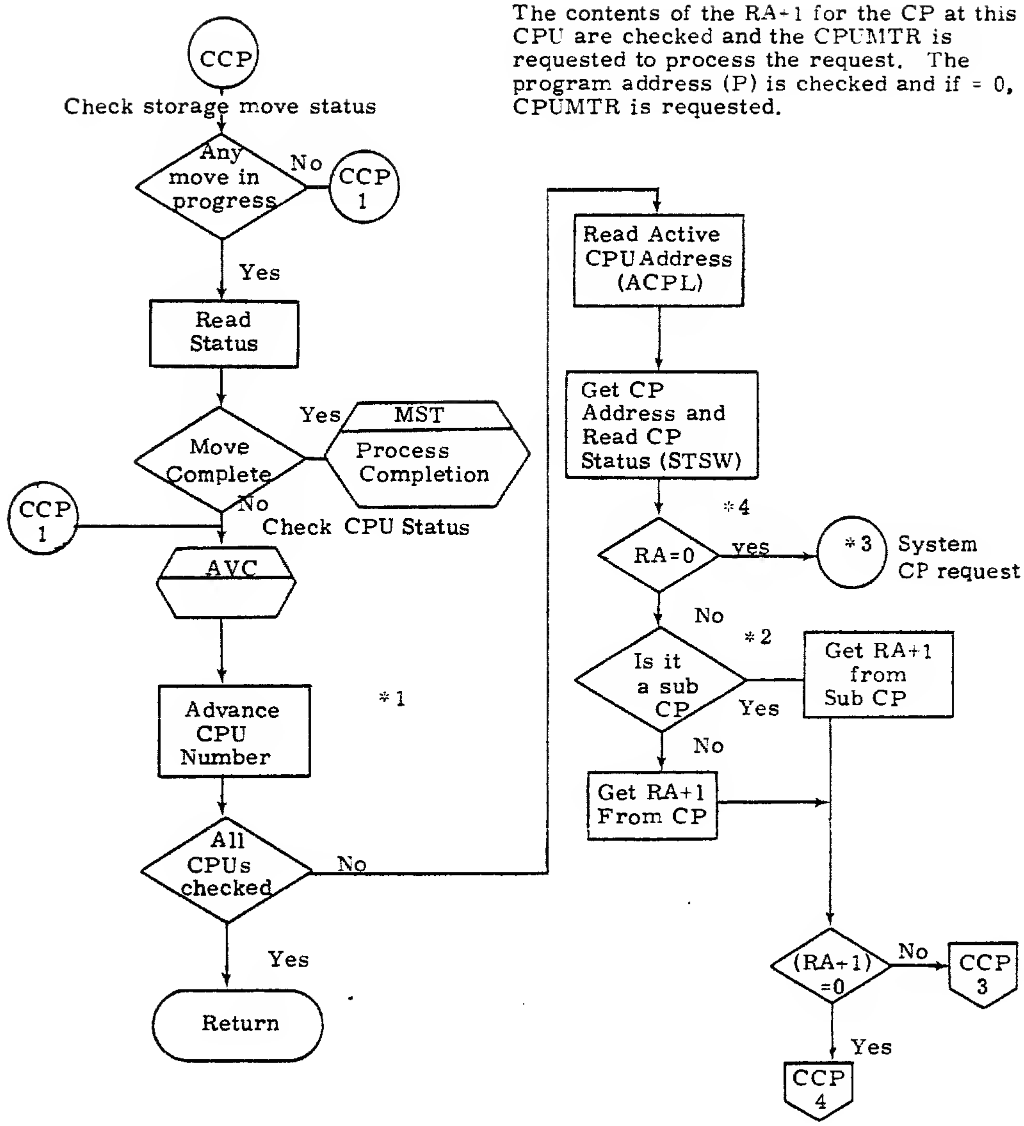
ENTRY: (A) = Function Number
(CM+1, . . . , CM+4) = Parameters
(CP) = CTR. pt. Area Address

EXIT: (cm, . . . , CM-4) = Response

CM is FTN request word.
CN is CPR request word.

Figure 3-35. FTN — Process Monitor Function

The contents of the RA+1 for the CP at this CPU are checked and the CPUMTR is requested to process the request. The program address (P) is checked and if = 0, CPUMTR is requested.



*1 Check active CP in CPU0 then CPU1 get CP number in CPU0.

*2 A user CP is running, i.e., this is not CPUMTR.

*3 If CEJ/MEJ available, then go to CCP1, if not then go to XJ1 on next page.

*4 If (RA) = 0, then this is CPUMTR and we can ignore it.

Figure 3-36. CCP – Check Central Program

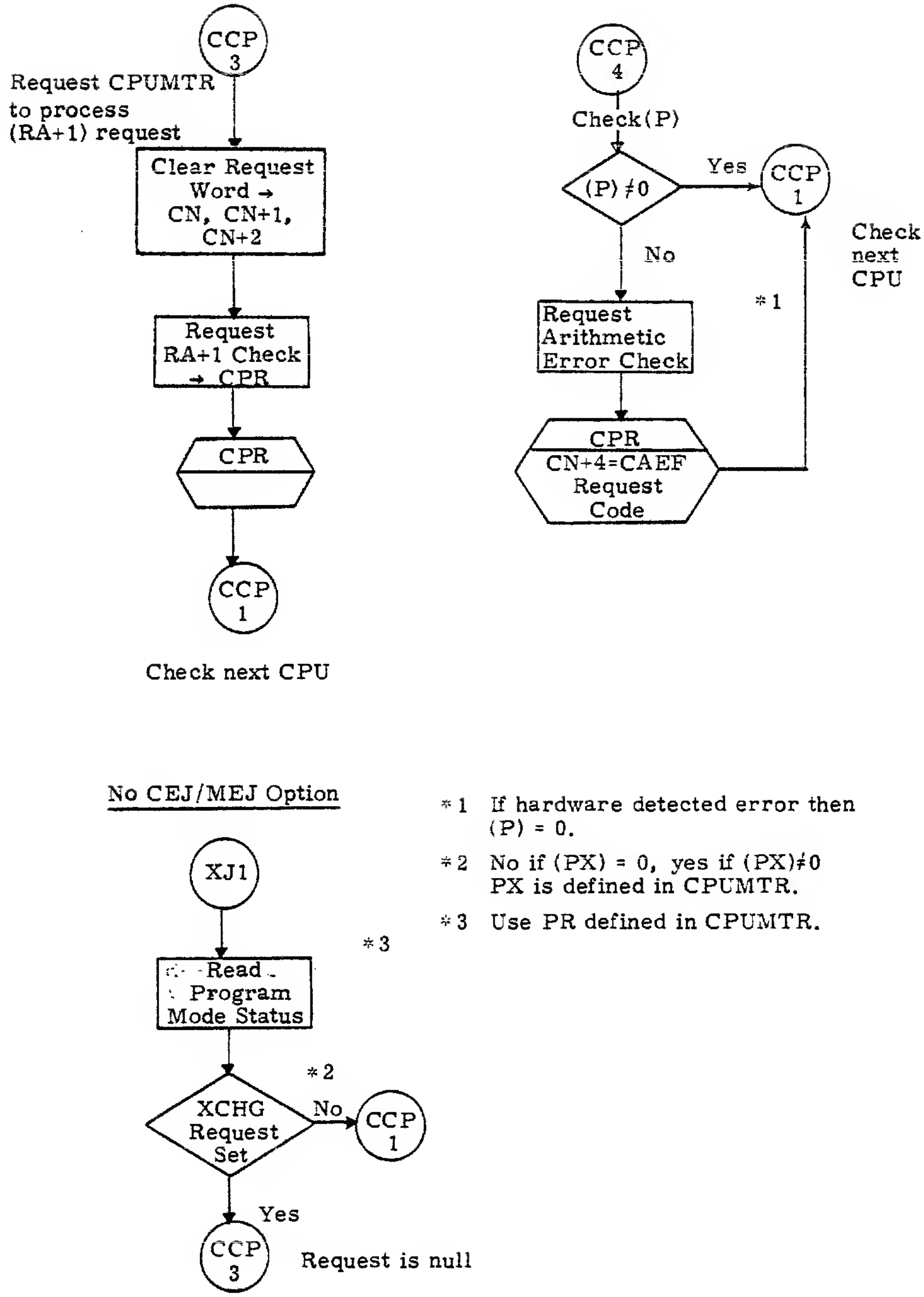
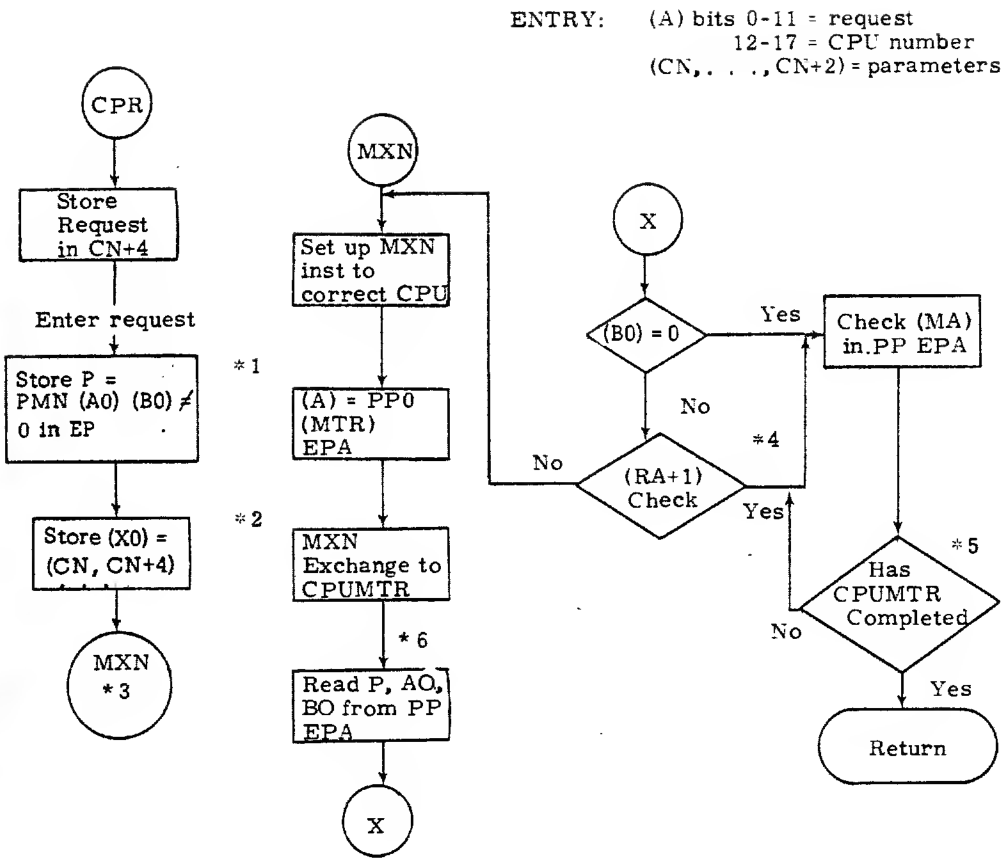


Figure 3-36. CCP – Check Central Program (Continued)



*1 This request will be processed by CPUMTR at PMN.

*2 PMN expects the request in X0.

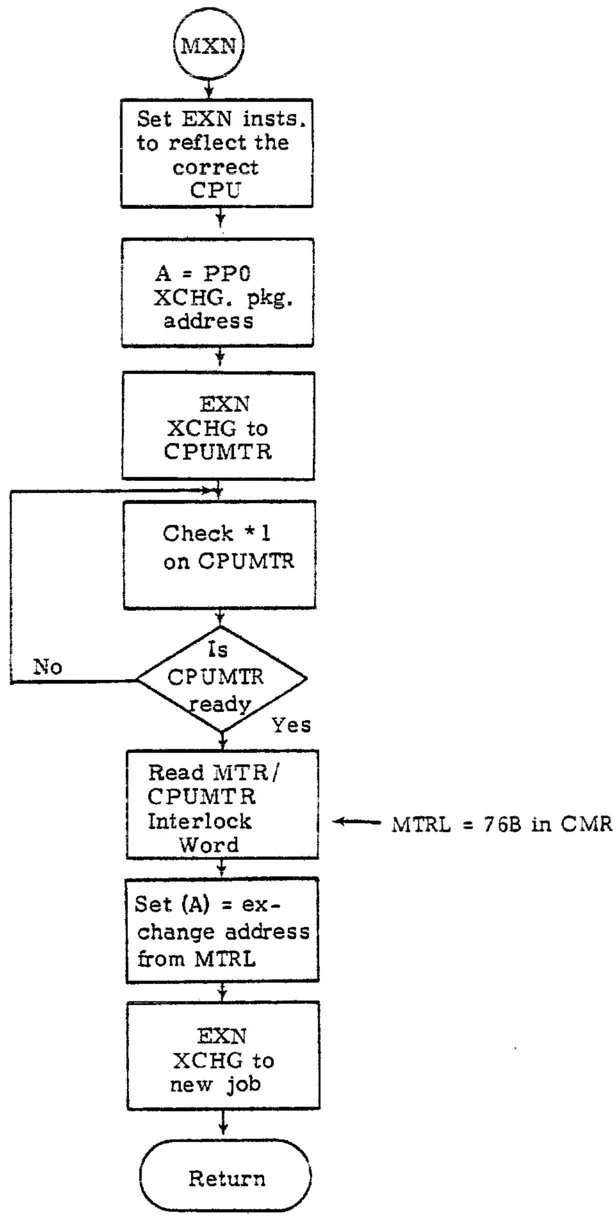
*3 If CEJ/MEJ option available, use code on this page.
If CEJ/MEJ option not available, use code of Figure 3-38.

*4 Was this an RA+1 check. If no and exchange occurred then CPUMTR is now running and he will automatically process this request. If not, then reissue the exchange.

*5 When CPUMTR has completed, he will exchange in the user CP interrupted whose EP is in PP0 EPA, as long as CPUMTR is executing CPn EP is in PP0EPA and any user (MA) = CPA of user CP. When CPUMTR exchanges to interrupted CP, then CPUMTR EP will be in PP0 EPA and CPUMTR (MA) = 0.

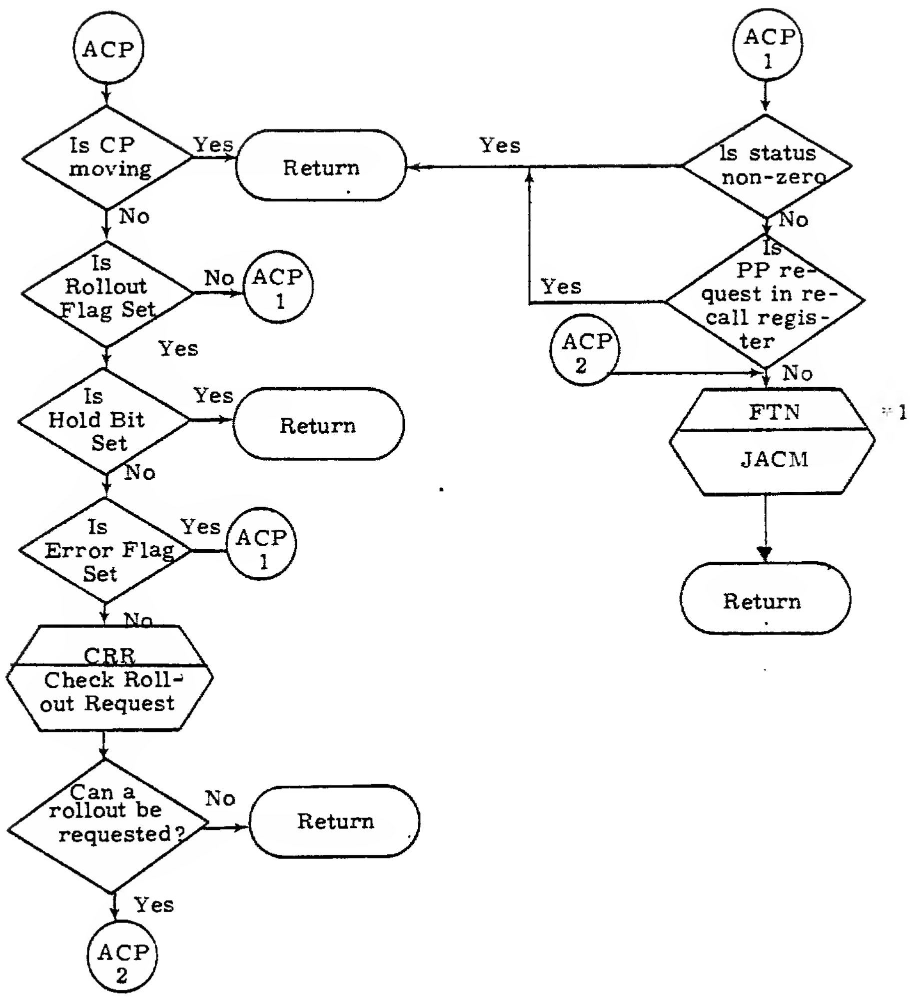
*6 There is a delay loop of 6 MICRO seconds.
LON 5
SBN 1
NJN *-1

Figure 3-37. CPR-CPUMTR Request Processor



*1 When CPUMTR has completed, it places the exchange address of the CP to be started in MTRL and jumps to a one word idle loop at CPSL = 77B in CMR, which is a zero word, i.e., a PS. MTR is doing an RPN 0 and waiting for (P) = CPSL.

Figure 3-38. XCHG - The CPU With CEJ/MEJ Not available



*1 Request job advancement.

Figure 3-39. ACP — Advance Control Point

3.7 CPUMTR – CENTRAL PROCESSOR MONITOR

CPUMTR is loaded in CMR and is entered at various places depending on what exchanged the CPUMTR.

The entry points are:

1. MTR – CPU program request
2. PMN – PP MTR request
3. PPR – Pool PP request
4. PRG – Program mode CPUMTR (system CP)
5. IDL and IDL1 – Idle packages

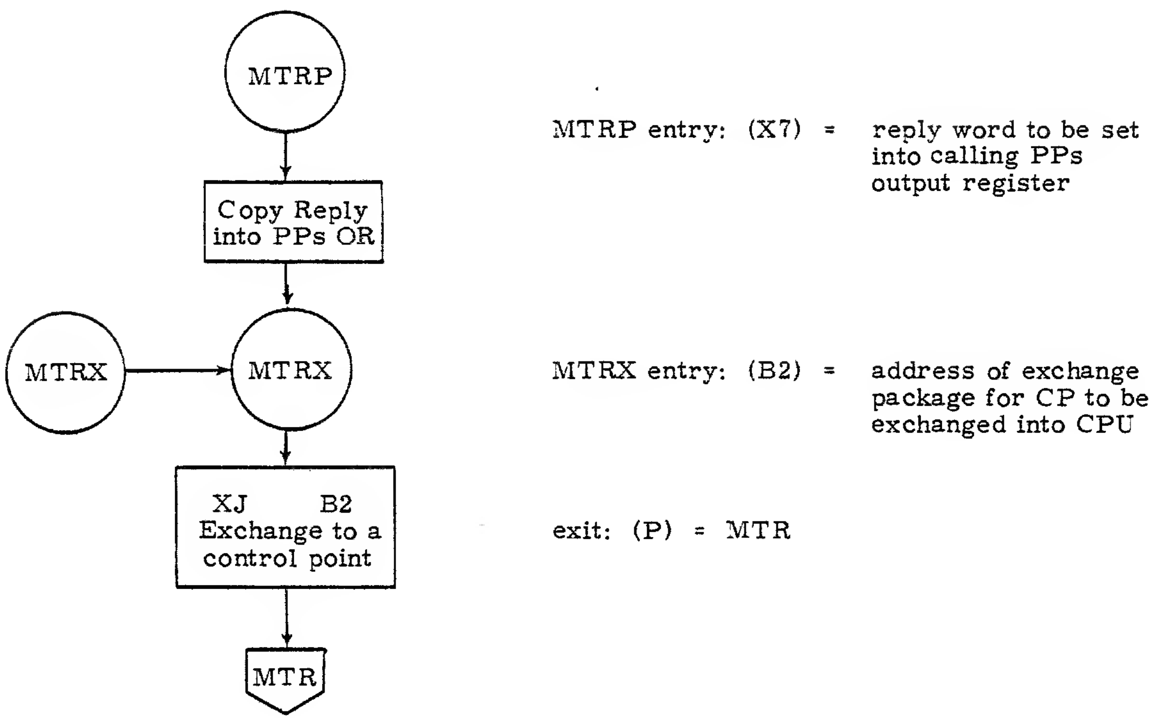
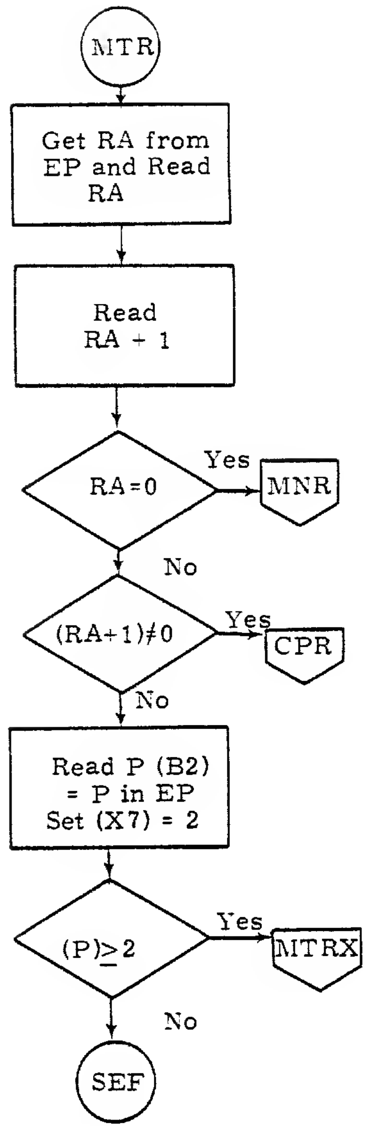


Figure 3-40. CPUMTR Return Points



Entry: (A0) = CPU number (0 or 1)
(B1) = 1
(B2) = address of caller's EP
(B7) = control point area address

NOTE: If CPn exchanged itself, then (B2) = (B7) and EP will be in CPA. If CPn was exchanged by MTR or some other pool PP, then (B2) = the address of the PP EPA which performed the exchange and (B7) = CPA.

Check for monitor request. Note that we are checking the address for RA against 0, i.e., is this exchange a CPUMTR EP or a CP EP.

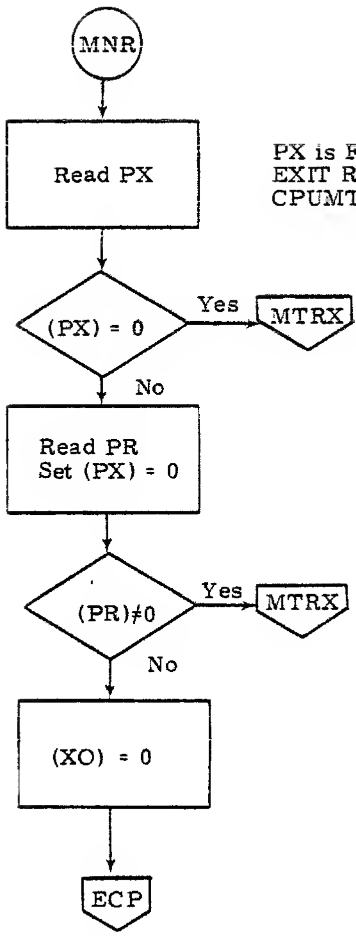
Process the RA+1 request.

Exchange CP back in. CP wanted a short pause.

Set error flag CPU detected on ARITH error. Uses (B7) = CPA, (X7) = error code. SEF will abort the CP program on ARITH error.

SEF is not flowcharted.

Figure 3-41. MTR – Exchange Entry From A CPU Program



PX is FWA of CPUMTR code and is the PROGRAM MODE EXIT REQUEST. It is set when program mode portion of CPUMTR has determined that this CP job is complete.

This is to determine if a CP or if the system CP (CPUMTR program mode) was interrupted.

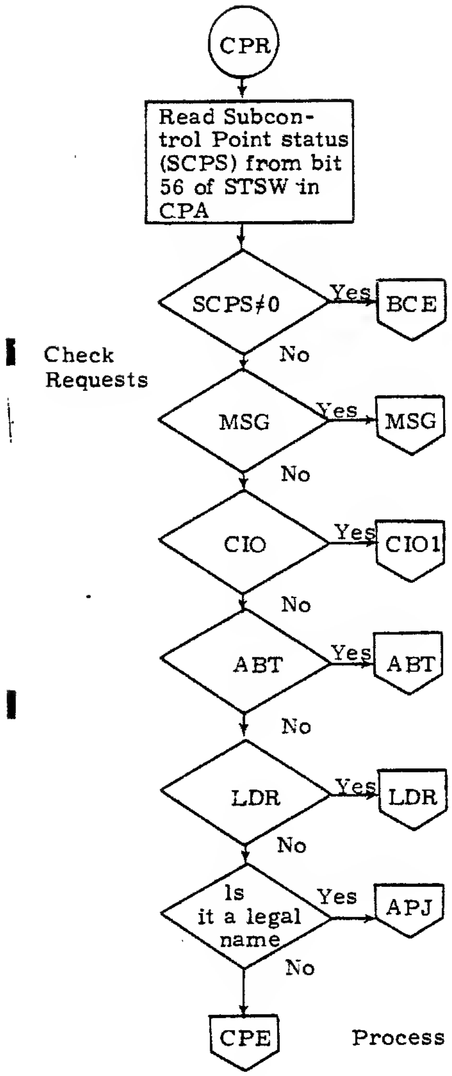
PR is a pointer to a stack of requests for program mode execution (i. e., system CP queue).

Let system CP continue running.

ECP end central program.
Uses (B7) = control point area address
(X0) = status bits
By setting (XO) = 0 we set CP status = 0 or not active.

ECP is not flowcharted.

Figure 3-42. CHECK – For System CP Request



Entry: (B3) = RA
(B7) = CPA
(X5) = (RA+1)
(A2) = address of RA in EP
(A5) = RA+1

None of these CP procedures are flowcharted.
If they end normally, they exit via MTRX.

BCE - begin Control Point Executive

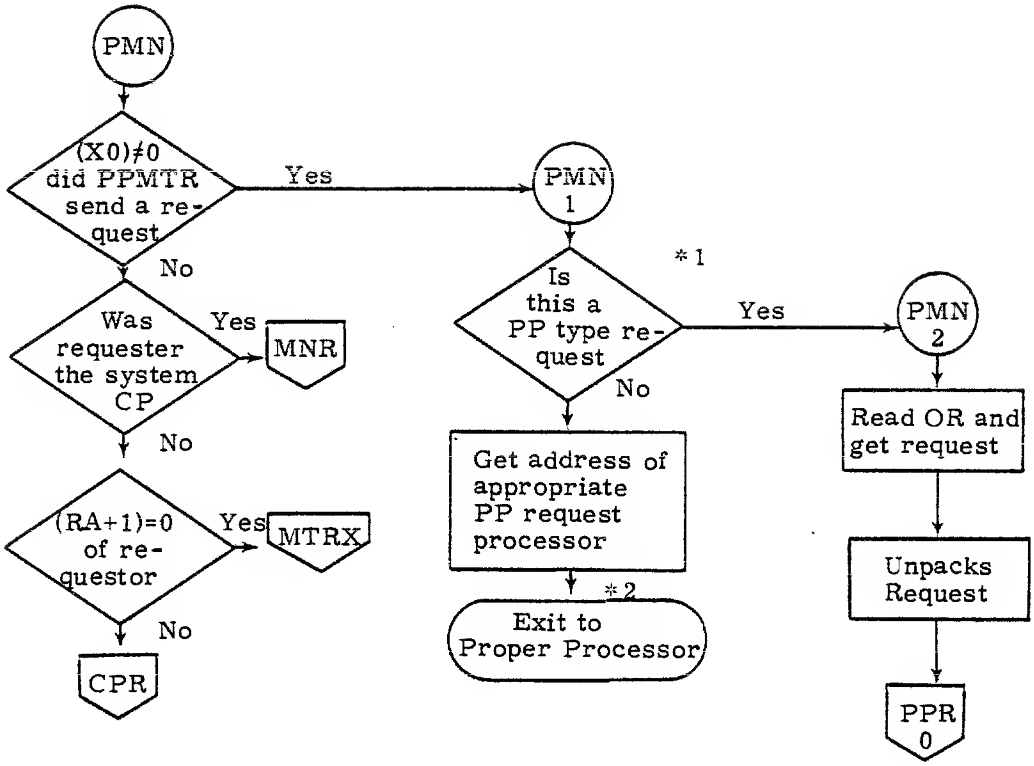
There are 9 more requests as of August 1, 1973. They are:

CPM
END
RCL
TIM
RSB
RFL
XJP
TLX
XJR
SIC

assign a PP put request in PPS IR.

Process CPU Call Error

Figure 3-43. Process — RA+1 Requests

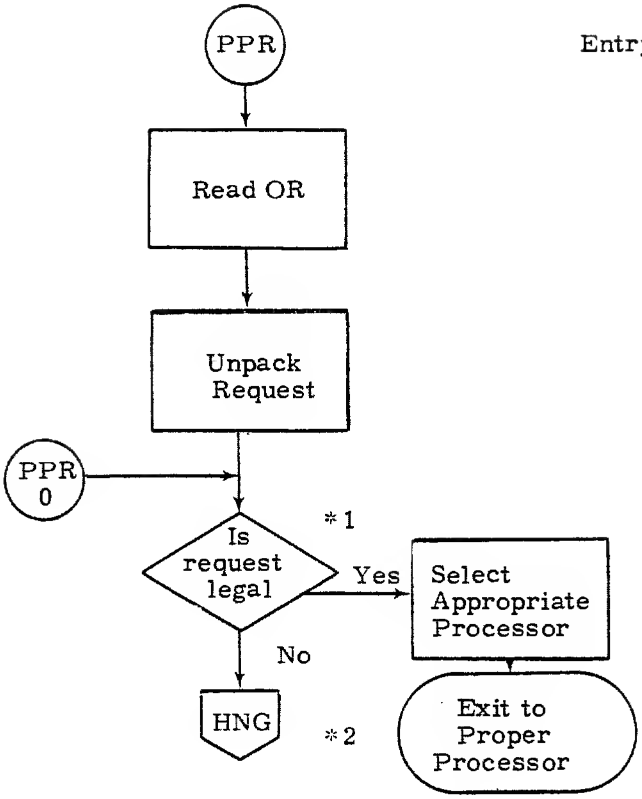


*1 MXPF is maximum number a PPMTR request can be, so test is (X0) - MXPF > 0, then go to PMN2.

*2 Those processors which require program mode CPUMTR will exit via EPR. EPR will check to see if the system CP was interrupted for this request and if so, will exit to MTRX. If a CPn was interrupted, then it will exit to BCP1, which will place this now deactivated CPn into "W" status, and then exit to MTRX.

Figure 3-44. PMN – Exchange Entry From PPMTR

Entry: (A5) = address of calling PPs OR
(B2) = address of calling PPs EP



Each processor will exit:
to MTRP with reply word in (X7) for PPUs OR if necessary,

- OR -

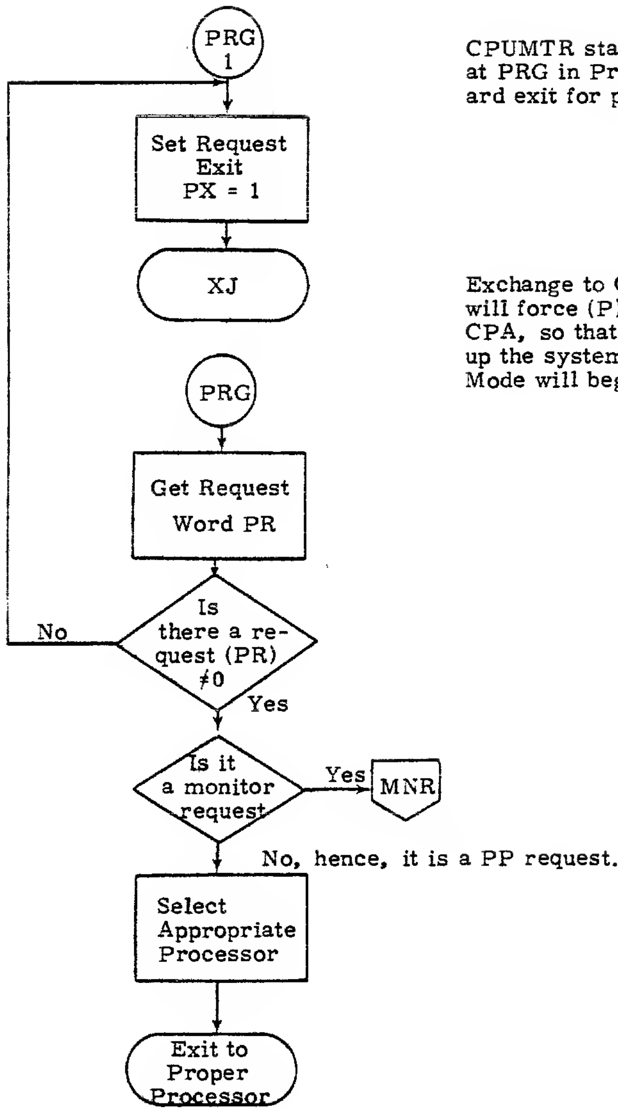
if no reply word necessary, then exit to MTRX.

If the processor requires program mode CPUMTR then the macro PPR will generate a queue entry and set up the EP, then jump to PRG will see no request and jump PRG1.

*1 Check to see if request (which is a number) is larger than the maximum.

*2 Hang PPU by not clearing OR, and display message PP HUNG at System CP.

Figure 3-45. PPR – Exchange Entry for Pool PPUs



CPUMTR starts the Program Mode portion at PRG in Program Mode. This is the standard exit for program mode CPUMTR.

Exchange to CPUMTR in monitor mode. This will force (P) = PRG in EP in the system CP CPA, so that the next time CPUMTR starts up the system CP, execution in Program Mode will begin at PRG.

Figure 3-46. PRG – Exchange Entry for System CP (Program Mode CPUMTR)

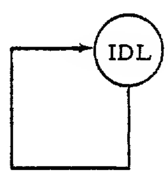
IDL and IDL1 – CPU0 and CPU1 Idle Loops

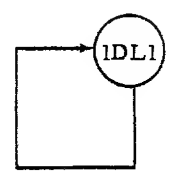
The exchange packages for IDL and IDL1 are loaded at the end of CPUMTR

```
(P)   = 2
(RA)  = location of IDL in CPUMTR
(FL)  = 5
(MA)  = location of this EP
(EM)  = 0
all other registers = 0
```

```
(P)   = 2
(RA)  = location of IDL1 in CPUMTR
(FL)  = 5
(MA)  = location of this EP
(EM)  = 0
all other registers  =  0
```

		Program IDL				Program IDL1	
0000	IDL	CON	0	(RA) for idle routines	IDL1	CON	0
0001		CON	0	(RA+1=0) for idle routine		CON	0
				never any requests			
0002		EQ	2	jump to itself		EQ	2





Program IDL and IDL1 will run until a PP or MTR interrupts them and exchanges CPUMTR into the CPU. If CPUMTR finds no other jobs to run he will exchange IDL or IDL1 back into the CPU.

4.0 PPR SYSTEM INTERACTION

Peripheral Processor Resident (PPR) provides the communication links between the PPs and the CPs. It serves as a PP idle program, the loader of PP programs and routines, and a convenient source of commonly used subroutines for other programs and routines. PP Resident is loaded at dead start time by STL and is never changed.

Each PP is a separate entity which can function independently of the CPU and operating system. It is desirable for the PPs to function in conjunction with and as a servant to the operating system. Therefore, PPR is designed to enable the PP to communicate with and work for the system.

When the system desires to start a PP program, CPUMTR will find an available PP (one whose IR = 0) and place the PP routine name (3 characters) and up to two arguments in the PPs IR. (i.e., up to 36 bits of arguments; see SYSTEM macro).

PPR consists of a set of routines shown in the core layout of Table 4-1.

Refer to Figure 4-1 for the following discussion of the flow of system to PP communications PPR reads its IR and if zero, will wait 128 microseconds through the Idle Loop before rereading. When the IR is not zero, PPR will request PLL using SLT to locate the requested routine which is in either the RPL or PLD. If the requested routine is not found, SFP will be loaded (refer to paragraph 4-9). If the requested routine is found, it is loaded by PLL and execution will then begin. As the routine is executing it can communicate with the system by issuing monitor requests via the FTN part of PPR (refer to Tables 3-1 and 3-2). FTN communicates with the system by placing the monitor function request in its OR and starting monitor.

Control is returned to the routine after the function has been processed.

CP programs can be moved when the PP pauses for relocation via PRL. When the routine uses an I/O device it reserves and releases channels via RCH and DCH. When the routine issues a message to the dayfile it uses DFM. When the routine loads and/or executes overlays it uses EXR. And, finally, when the routine drives an RMS device the SMS routine will load the proper driver and via entry points POS, RDS, and WDS can read and write. When the routine is completed it simply issues monitor function DPPM via FTN and jumps to PPR.

TABLE 4-1. POOL PROCESSOR MEMORY MAP (as of Release 8/24/73)

First word address	
0000	DIRECT CELLS
0100	PP RESIDENT (PPR) idle loop of PP
0125	PERIPHERAL LIBRARY LOADER (PLL)
0307	SEARCH LIBRARY TABLE (SLT)
0350	PROCESS PP MONITOR FUNCTION (FTN)
0414	PAUSE FOR RELOCATION (PRL)
0436	RESERVE CHANNEL (RCH)
0445	RELEASE CHANNEL (DCH)
0454	SEND DAYFILE MESSAGE (DFM)
0533	EXECUTE ROUTINE (EXR)
0544	SET MASS STORAGE (SMS)
0600	MASS STORAGE DRIVERS
1073	LOADER TABLE OF CURRENT PPU ROUTINE
1100	FWA of current PPU ROUTINE
7000 BUF	MASS STORAGE BUFFER 2 words of PRU control and 500 words of information. 502 words total
7502	MASS STORAGE ERROR PROCESSOR
7777	LAST WORD OF PPU

4.0.1 KRONOS/NOS PP Naming Conventions

NAME	DESCRIPTION
Yxx	RA + 1 CP callable routine
0xx	Location free routine
1xx	First level system routine callable by RA+1 TLX request
2xx	Second level overlay
3xx	Third level overlay
4xx	Fourth level overlay
5xx	Fifth level overlay
6zz	MS driver routine
7zz	MS error processor routine

NAME	DESCRIPTION
8xx	Unused
9xx	Syntax type used by DSD, DIS, 1LS, 1TD, and QIS

Where Y is any alpha character

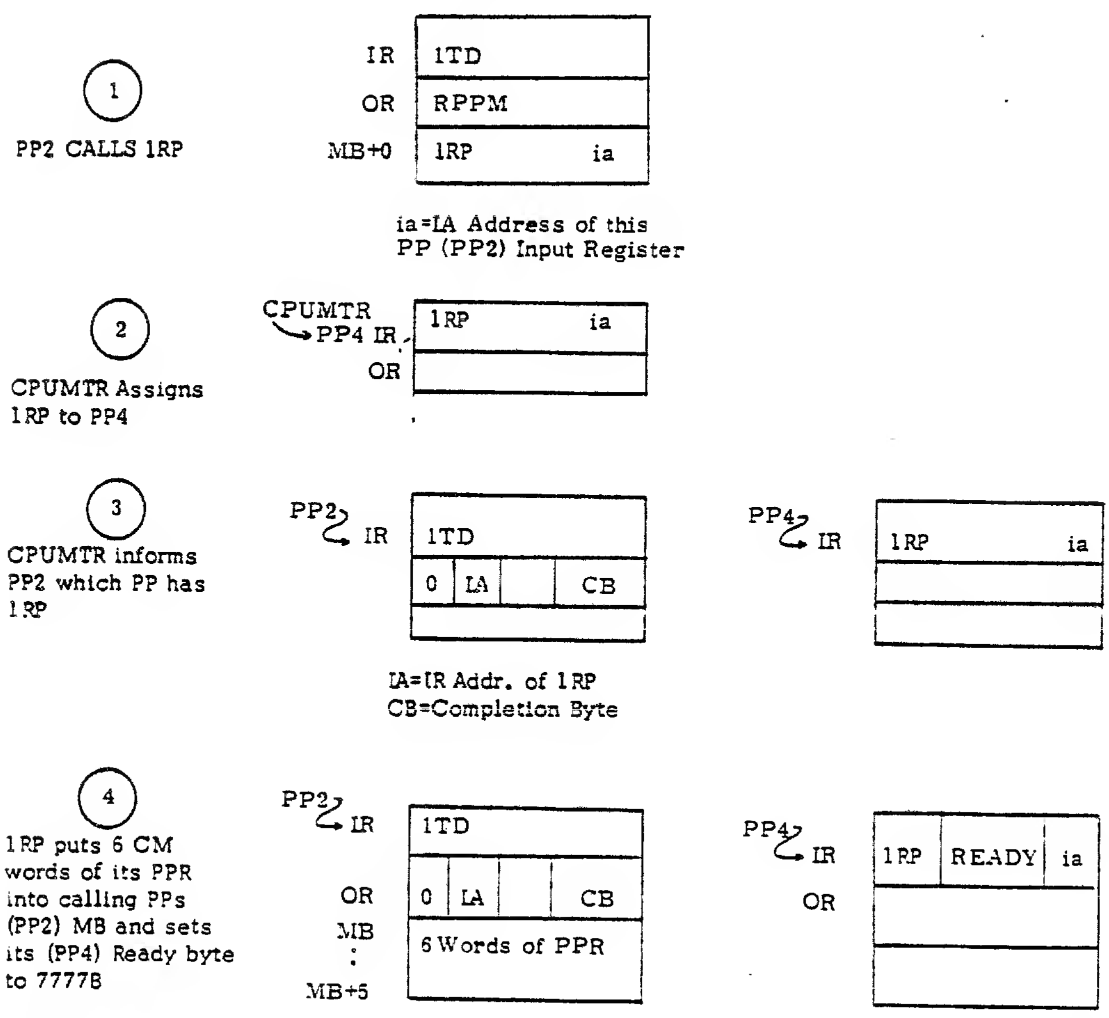
xx is any two characters including specials

zz is any legal MS mnemonic

4.1 PP RESIDENT

RESTORE PPR IF DESTROYED
(e.g., BY 1TD)

PP2 RESTORES PPR



Subsequently, PP2 restores 6 central memory (CM) words (36 PP bytes) of PPR and if 1RP has the ready byte set PP2 will set its completion byte (CB) $\neq 0$.

PP4 (1RP) will copy the next 6 central memory words (36 PP bytes) of PPR and set its ready byte.

1RP (PP4) and the requesting PP (PP2) continue until all of PPR has been transferred, then 1RP (PP4) drops.

1TD must set the Ready byte in 1RPs IR within 1 sec upon regaining control or 1RP will drop.

1TD will set CB to zero when it is ready to receive 36 bytes of PPR.

1RP will set CB to 400X where X is the number of CM words transferred by 1RP. When 1RP is done it will set CM to 7777 and drop.

4.2 DAYFILE MESSAGES (ALL MESSAGES ARE ISSUED BY SFP, SPECIAL FUNCTION PROCESSOR)

The dayfile messages are:

"XXX NOT IN PP LIB." = PP Package XXX was not found in PP LIBRARIES.

"XXX NOT IN PP LIB. - CALLED BY YYY." = PP Package XXX was not found in the PP LIBRARIES and was called by package YYY.

"SFP/XXX PARAMETER ERROR." = Parameter address outside FL.

SFP/XXX ILLEGAL ORIGIN CODE." = Function Illegal for users job origin.

"SFP CALL ERROR." = SFP not loaded by default.

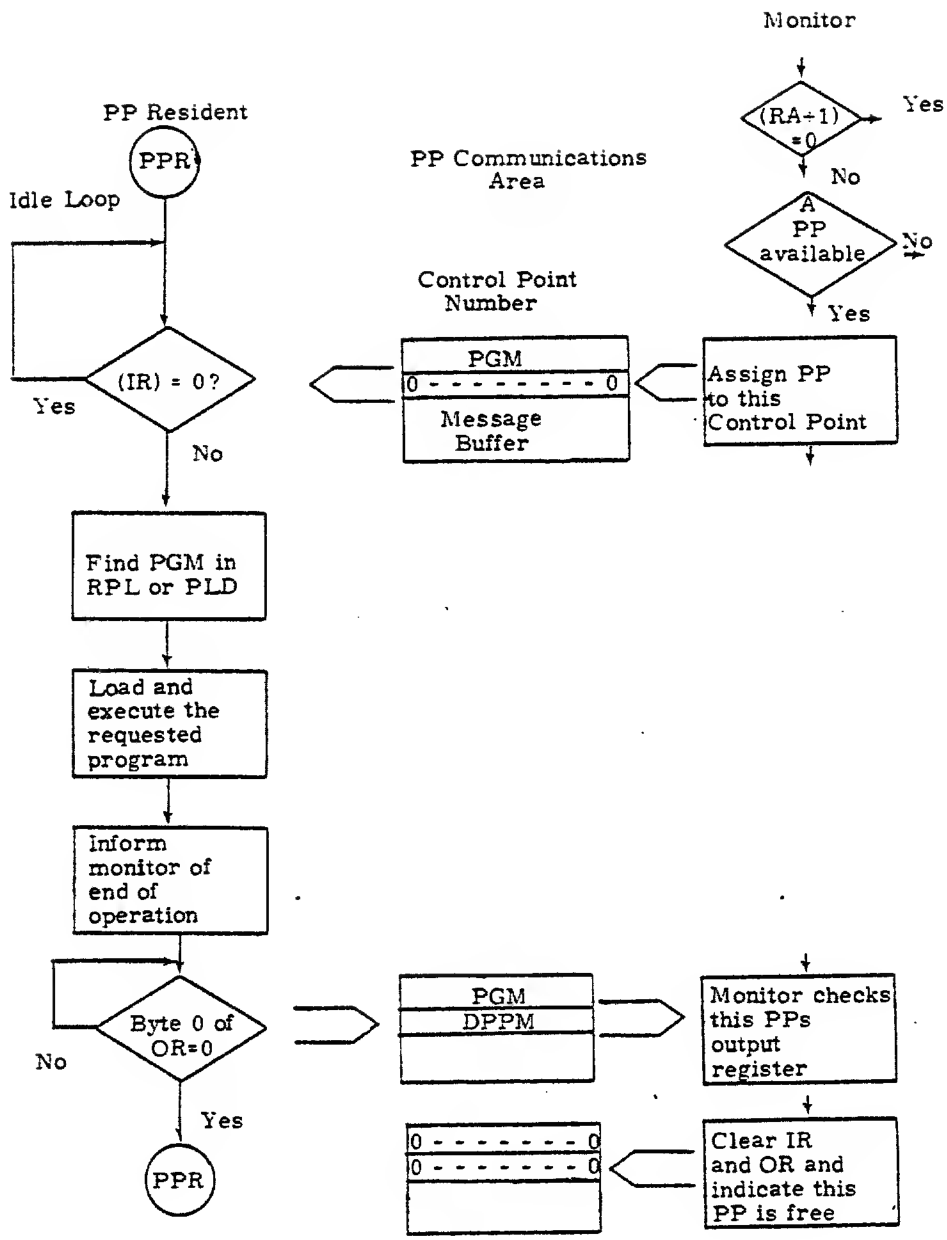


Figure 4-1. System Interaction – PPR

4.3 ROUTINES USED

The following routines are used:

1DD - PROCESS DAYFILE DUMP.

A MASS STORAGE DRIVER.

SFP - SPECIAL FUNCTION PROCESSOR

Used for SCOPE 3.4 compatibility and is called if PLL does not find an entry in the PLD.

The following routines must reside in RPL: 1DD, SFP, LSL, 0DF and all the mass storage drivers.

The four instructions CRM, CWM, IAM, and OAM use cell T0 to hold the value of the P register while executing. If T0 is changed during execution of any of these instructions, the next instruction to execute will be at the location specified in (T0). This is used by SFP to autoload and force an immediate transfer to the instructions just read in.

Table 4-2 shows the Direct Location Assignments available for PP programs.

The direct cells ON, HN, TH, TR, IA, OA, and MA are the only cells that must not be changed by a user PP routine.

4.4 PP RESIDENT INITIALIZATION

The PP resident initialization routine PRS is initiated at dead start and will be overlayed by the first mass storage driver loaded (i.e., PRS resides at Location 600). PRS prepares the PP Resident routine by setting up the proper cells and setting up the exchange package address in the routine FTN. The procedure is described in Section 24 on Deadstart.

4.5 LOADING PP ROUTINES

The CPUMTR, or any PP routine, can place the name of a PP routine in an input register. The CPUMTR will determine the availability of PPs and will pick the next available PP for a PP program request. If a PP program is told by CPUMTR that no PPs are available, he may load the requested routine by inserting the program name into this PPs own input register.

PPR loops on its input register checking for non-zero. When PPR finds it non-zero, he transfers to PLL to load the requested routine.

PLL will search the RPL (via subroutine SLT) for the name of the requested PP routine. If the name is not found, it will search the PLD. If the name is still not found, the last word of the PLD will force the loading of SFP (the Special Function Processor). SFP will determine if the

TABLE 4-2. DIRECT LOCATION ASSIGNMENTS

Symbol Name	Location	Description
T0	0	TEMPORARY STORAGE
T1	1	
T2	2	
T3	3	
T4	4	
T5	5	
T6	6	
T7	7	
CM	10	CM WORD BUFFER (5 LOCATIONS)
LA	15	PACKAGE LOAD ADDRESS
	SET BY PP RESIDENT BEFORE ENTRY TO PROGRAM.	
IR	50	INPUT REGISTER (5 LOCATIONS)
RA	55	REFERENCE ADDRESS/100
FL	56	FIELD LENGTH/100
	READ ONLY CONSTANTS.	
ON	70	CONSTANT 1B
HN	71	CONSTANT 100B
TH	72	CONSTANT 1000B
TR	73	CONSTANT 3B
	SET BY PP RESIDENT BEFORE ENTRY TO PROGRAM.	
CP	74	CONTROL POINT ADDRESS
	READ ONLY CONSTANTS.	
IA	75	INPUT REGISTER ADDRESS
OA	76	OUTPUT REGISTER ADDRESS
MA	77	MESSAGE BUFFER ADDRESS
	PP RESIDENT ENTRY POINTS. "PPR" IS ENTERED BY A LONG JUMP. ALL OTHER ENTRY POINTS ARE ENTERED BY A RETURN JUMP.	
PPR	103	PP RESIDENT IDLE LOOP
PLL	125	PP LIBRARY LOADER
FTN	364	PROCESS MONITOR FUNCTION
PRL	424	PAUSE FOR RELOCATION
RCH	437	RESERVE CHANNEL
DCH	446	RELEASE CHANNEL
DFM	501	PROCESS DAYFILE MESSAGE
EXR	533	EXECUTE ROUTINE
SMS	547	SET MASS STORAGE
	OTHER CONSTANTS	
PPFW	1100	FIRST WORD ADDRESS FOR PP PROGRAMS
ESTS	551	CONTAINS FIRST WORD ADDRESS OF EST

call is for one of its functions. If so, SFP will check the parameters if they exist. If any part of the call is incorrect or if the PP routine requested is not one of its functions, it will issue the appropriate dayfile message, inform CPUMTR, clear thê input register, and return control to PPR.

If, at any time the requested PP routine is found, it will be loaded and control will transfer to the first instruction at PPFW.

4.6 6000-SERIES PP ABSOLUTE CODING FORMAT

Binary output for a 6000-Series PP program or overlay is a logical record that may contain the following:

A prefix table

A 6000-series PP program control table.

The PPU text in five PPU words per 60-bit CPU word.

The format of the control table is:

59	41	35	23	11 0
Name	00	fwa	0000	length

Bits	Field	Description
59-42	name	Program name, 1-to-3 display code characters, left-justified with zero fill.
41-36	none	Reserved for future system use.
35-24	fwa	Origin-5; address at which header word is loaded.
23-12	none	Reserved for future system use.
11-0	length	Number of CM words in program image (1/5 the number of PPU words rounded up).

4.7 CM TABLES USED BY PP RESIDENT

PP Resident uses the following CM Tables:

1. PP Communication Area

	18	1	5	36
IR	3-character PP routine name	1	C.P. #	Parameters

	12	48
OR	Monitor Function Code	Parameters

MB 6 CM words

2. Resident Peripheral Library (RPL)

	18	6	12	12	12
RPL+0	program name 1	0	Load Address	0	Length
	first word	of	binary deck		
i	program name n	0	Load address	0	Length
n	0	0	0	0	0

indicates end of library

3. Peripheral Library Directory (PLD)

	18	6	12	12	12
PLD+0	program name 1	0	Load address	track	sector
1	program name 2		Load address	track	sector
	program name 3		Load address	track	sector
	program name 4		Load address	track	sector
i	program name n		Load address	track	sector
n	0		RPLA	Length of SFP	LA of SFP

The last word of the PLD is a dummy entry. It forces PLL to load SFP.

RPLA is the address in RPL of the routine SFP

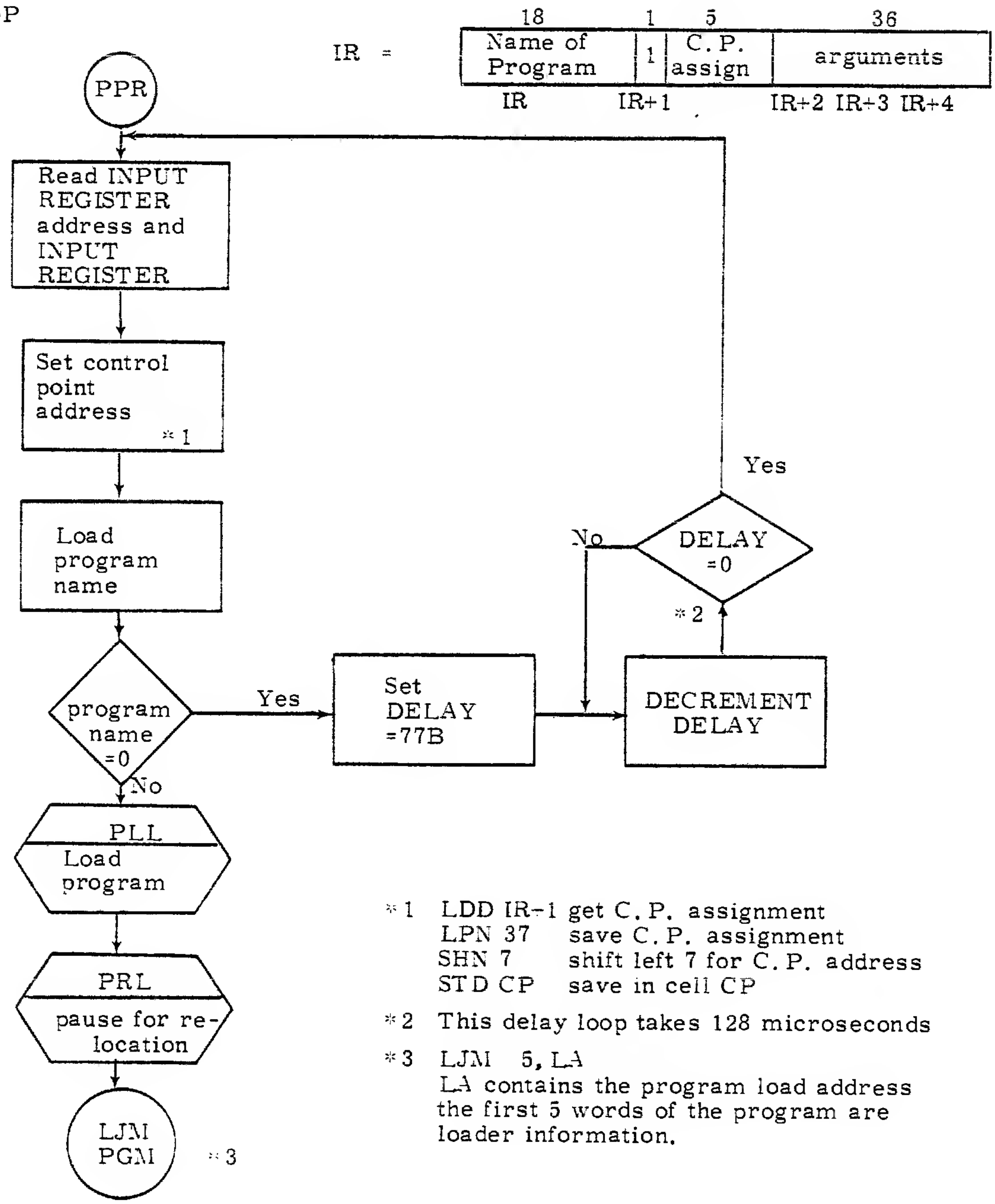
Length of SFP in CM words in RPL

LA of SFP is Load Address of SFP

4.8 PP RESIDENT ROUTINE FLOWCHARTS

The following flowcharts illustrate the PP Resident routines.

IDLE LOOP



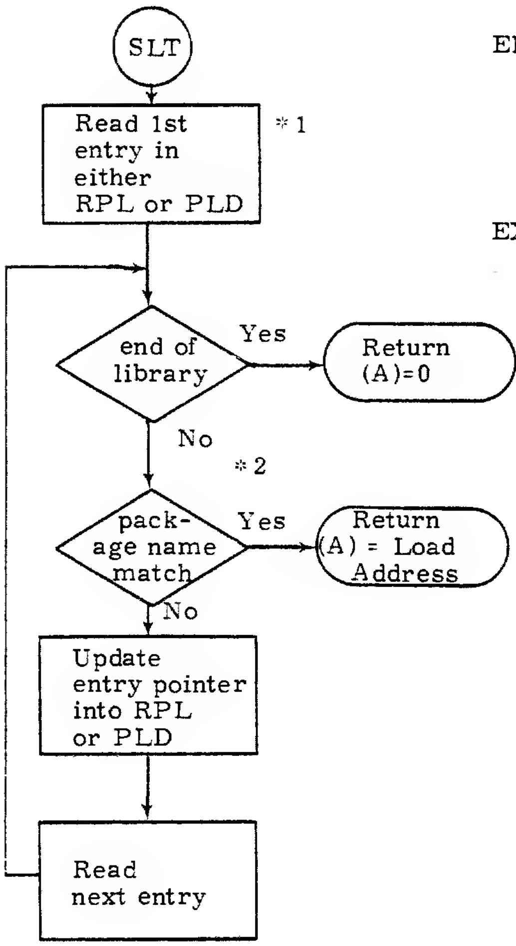
*1 LDD IR+1 get C.P. assignment
LPN 37 save C.P. assignment
SHN 7 shift left 7 for C.P. address
STD CP save in cell CP

*2 This delay loop takes 128 microseconds

*3 LJM 5, LA
LA contains the program load address the first 5 words of the program are loader information.

Figure 4-2. PP Resident (PPR)

RPL =	Program name	0	Load address	0	CM word length
PLD =	Program name	0*3	Load Address	track	sector
Direct Cell	CM	CM + 1	CM + 2	CM + 3	CM + 4



ENTRY (A) bits 0-11 = address advance instruction
(A) bits 12-17 = library pointer address
part of SLTB + SLTB + 1 = package name *1
(LA) = Load address for zero overlays

EXIT (A) = Load address if package found
= 0 if package not found
(PLLA) = Load address if package found
(T1-T2) = Address of package entry

*1 Use initialization instructions in (A) in order to read either RPL or PLD.

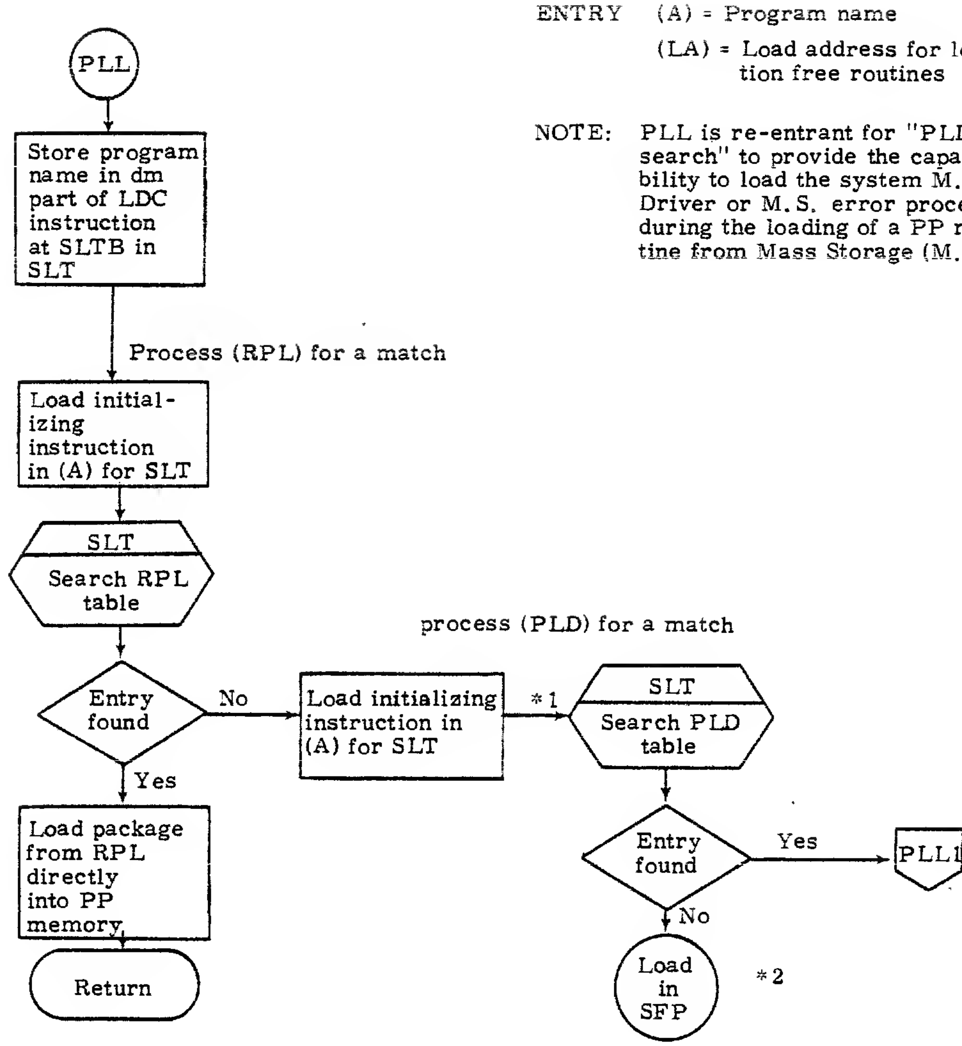
*2 The package name was stored into the compare instruction at SLTB by PLL
SLTB LMC *

*3 ASR Equipment.

Figure 4-3. SEARCH LIBRARY TABLE SLT

ENTRY (A) = Program name
(LA) = Load address for location free routines

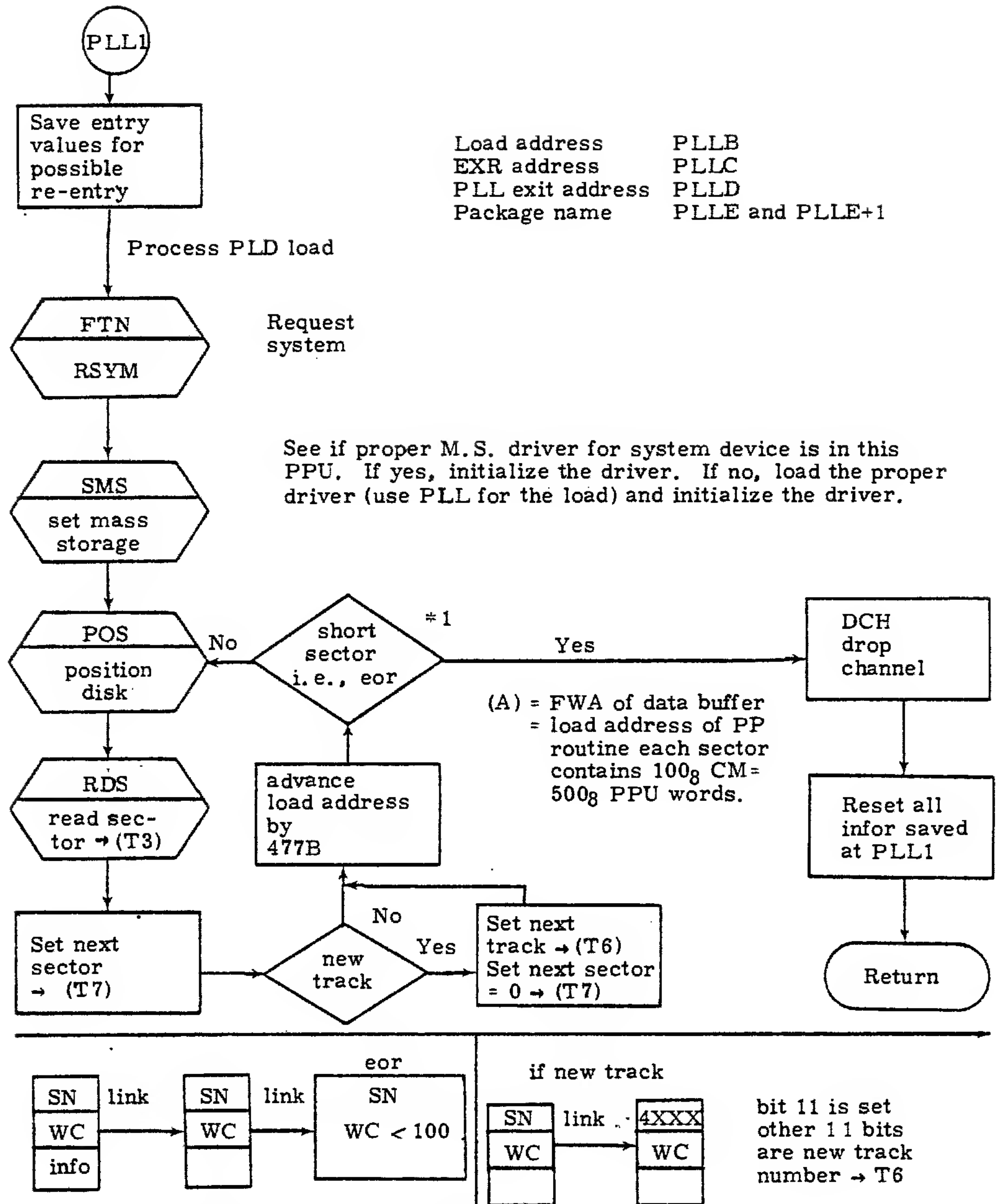
NOTE: PLL is re-entrant for "PLD search" to provide the capability to load the system M.S. Driver or M.S. error processor during the loading of a PP routine from Mass Storage (M.S.).



*1 SLT uses these instructions to point at the proper table and to increment thru them.

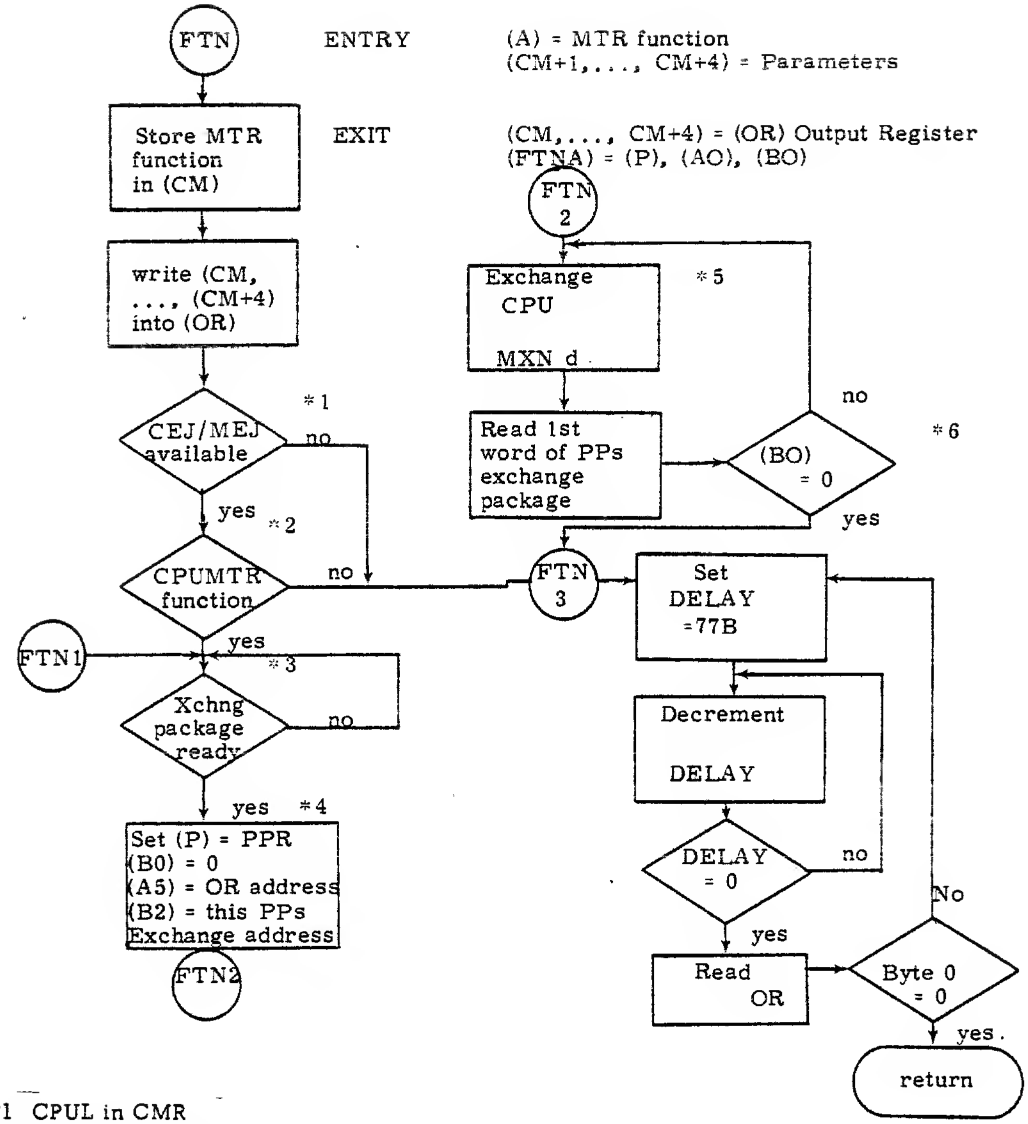
*2 See SFP on page 4-22

Figure 4-4. Peripheral Library Loader (PLL)



SN = Sector number, WC = Word count
*1 Example of program on RMS.

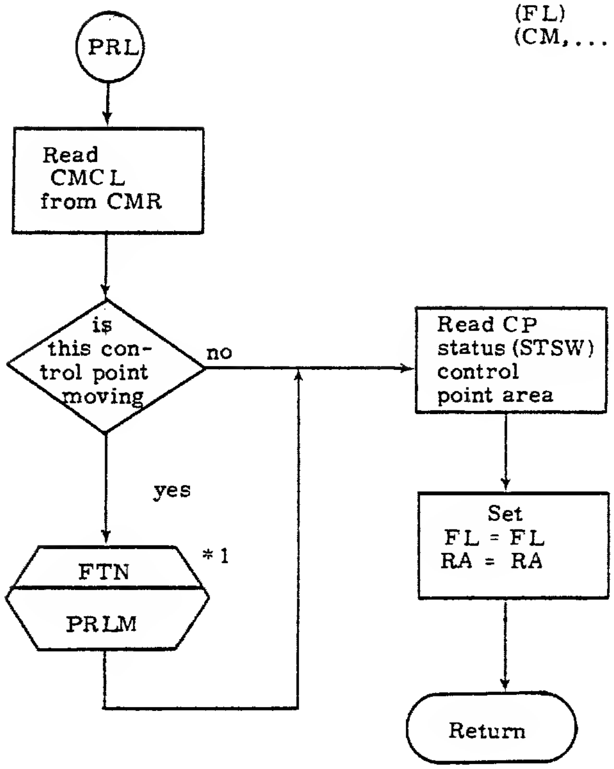
Figure 4-4. Peripheral Library Loader (PLL) (Continued)



*1 CPUL in CMR
*2 is function < 36B
*3 check (MA) for zero see Figure 2-4.
*4 exchange area in CMR (BO) (AO) and (P) are from PXPP+1 in CMR which is stored into 5 local cells in FTN. BO is set = 1477B so FTN can check if the exchange was honored. from D/S PRS; P VFD 24/PPR; AO VFD 18/0; BO VFD 6/0; LON 77 = VFD 12/1477B; hence BO = 18/001477B. See *6
*5 activate CPUMTR
*6 has CPUMTR been activated

Figure 4-5. Process MTR Function (FTN)

ENTRY	(CP) = Control Point Address	
EXIT	(A)	= RA
	(RA)	= RA
	(FL)	= FL
	(CM, . . . , CM+4)	= CP status word (STSW)

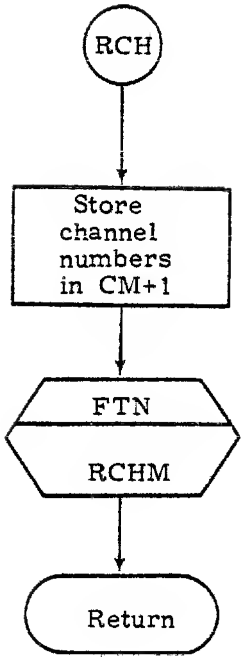


*1 Pause for Storage Relocation

Figure 4-6. Pause for Storage Relocation (PRL)

ENTRY (A) = 1 or 2 channel numbers *1
(CM+2) = additional channel numbers (if more than 2 needed)

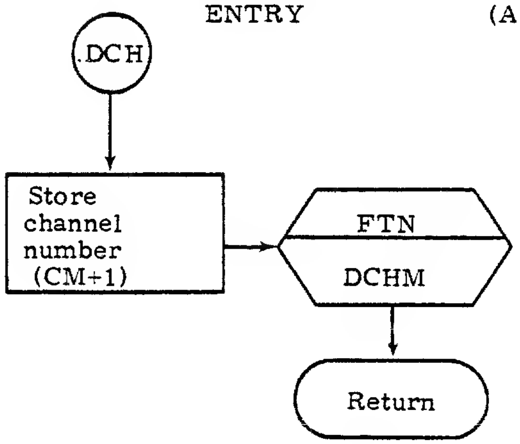
EXIT (CM+1) = assigned channel



Reserve channel function

RELEASE CHANNEL (DCH)

ENTRY (A) = channel number



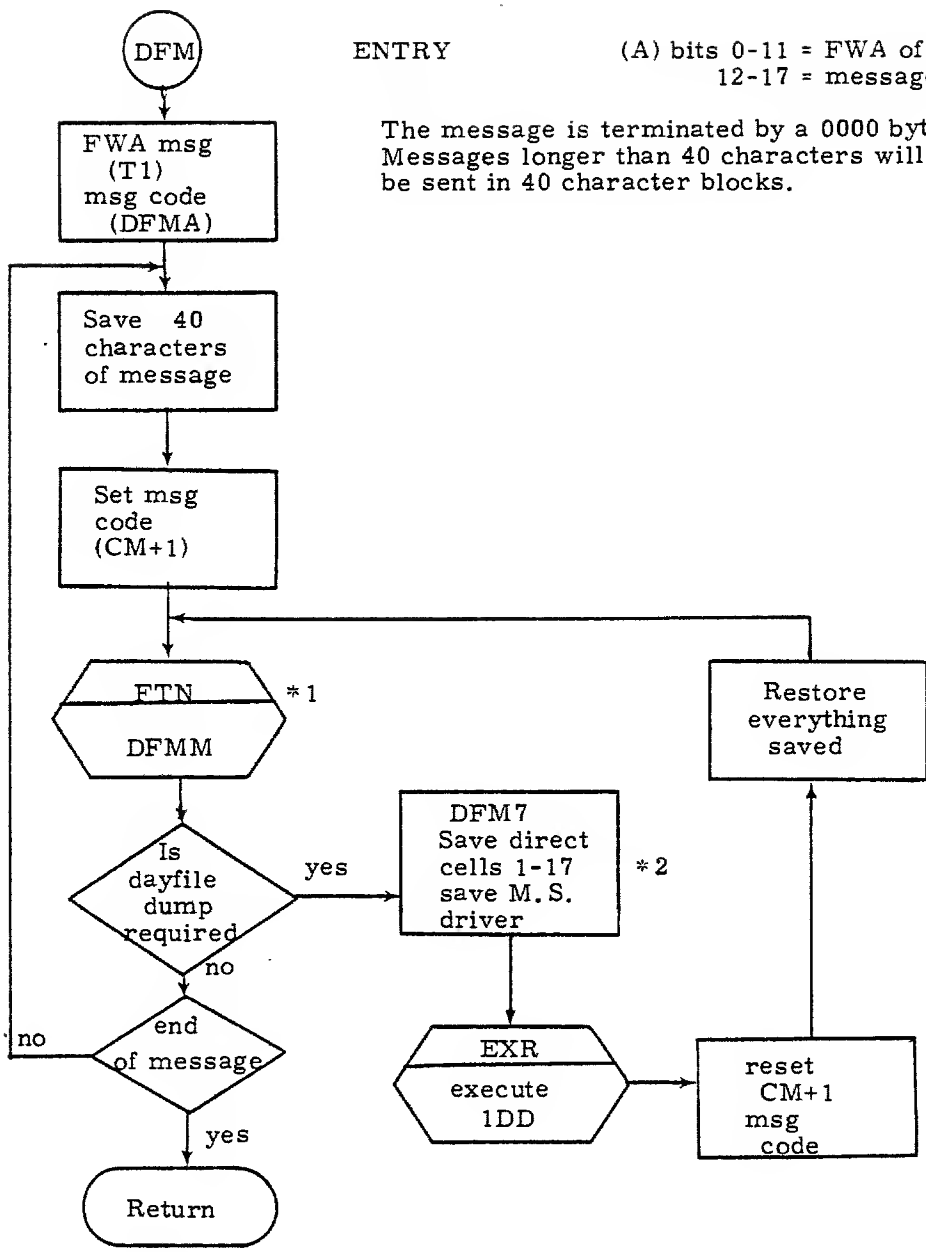
Release channel function

*1 RCHM will assign one of the channels requested if it can. (A) and (CM+2) are used for optional channels

Figure 4-7. Reserve Channel (RCH)

ENTRY (A) bits 0-11 = FWA of message
12-17 = message control word

The message is terminated by a 0000 byte. Messages longer than 40 characters will be sent in 40 character blocks.



*1 Dayfile message function
*2 1DD may use a different driver to dump the dayfile and the original M.S. driver must be reloaded for the user. If dayfile dump required, then enough PP bytes must be saved for the part of 1DD which is loaded above PPFW and destroys that portion of the calling routine. These bytes are saved in the dayfile dump buffer in CMR. CPUMTR assures only one PP will do a dayfile dump by not clearing the OR of a PP until the dayfile dump buffer is clear.

Figure 4-8. Send Dayfile Message (DFM)

Dayfile Message Options

Dayfile message options are:

A normal dayfile message is sent to the master dayfile, control point dayfile, and control point message area. The "job name" is defined in the control point area.

	-	00000	NORMAL MSG.
NMSN	-	10000	NORMAL MSG. WITH NO MSG AT CTL. PT.
JNMN	-	20000	MSG. TO MASTER DAYFILE ONLY, WITH JOBNAME
CPON	-	30000	MSG. TO CTL. PT. DAYFILE ONLY
ACFN	-	40000	MSG. TO ACCOUNT DAYFILE ONLY
AJNN	-	50000	MSG. TO ACCOUNT DAYFILE WITH JOBNAME
ERLN	-	60000	MSG. TO ERROR LOG ONLY
EJNN	-	70000	MSG. TO ERROR LOG ONLY WITH JOBNAME

ENTRY (A) = Routine name
(LA) = Load address for location free routines

EXIT Exit to called routine via simulated return jump from caller

Example: Call overlay 2XY

(A) = 2XY
(LA) = load address
RJM EXR

then core from (LA) to (LA) + 7 is

(LA)+0	2X
1	Y-
2	load address
3	0
4	length
5	0100 LJM
6	return address from caller of EXR
7	1st executable statement address

program 2XY at completion does a RETURN, which is a LJM (LA) + 5, which will LJM (return address from caller).

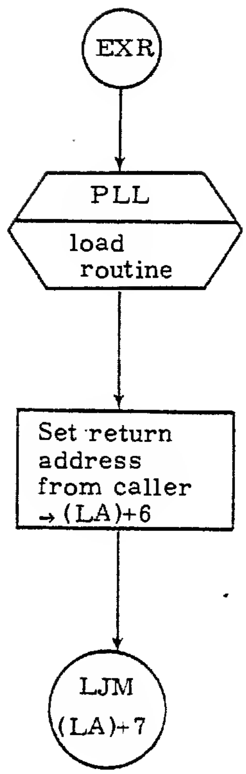


Figure 4-9. Execute Routine (EXR)

4.9 MASS STORAGE DRIVER RESIDENT AREA

Mass storage drivers are overlays loaded by PP Resident in an area between PP Resident and the first word address of PP programs. Mass storage drivers are coded such that the entry points remain constant between all drivers.

Parameters passed to the driver are:

(T4) = Channel
(T5) = Equipment number
(T6) = Track
(T7) = Sector

The rules are:

- Name = A "6" followed by the equipment mnemonic
- Origin = "MSD"
- First word = Driver identification, i.e., the last 2 characters of the driver name.
- Fourth word = A jump to the driver initialization routine. This entry is used by "SMS" to cause initialization of the driver. Exit from initialization is to "SMSX". "SMS" enters the initialization routine with (CM - CM+4) = EST entry.
- The entries for read, write, and position origined at the appropriate symbolic names. i.e., "RDS", "WDS", "POS". These entries are entered via return jump, and due to the definition of the names, transfer code is necessary.
- The number of sectors/track (2 numbers) origined at "SLM".
- The driver must not use any direct locations except T1, T2, CM - CM+4, LA.
- The driver and its associated error processor must reside in RPL.

All drivers are overlayed in this area, and use the following three entry points.

POS - Position disk.

Must be called on every track switch

Entry driver initialized (SMS called).

(T4) = Channel
(T5) = Equipment
(T6) = Track
(T7) = Sector

Exit None.

RDS - Read sector

Entry driver initialized (SMS called).

(T4) = Channel
(T5) = Equipment
(T6) = Track
(T7) = Sector
(A) = FWA of data buffer. (502 *1 word buffer needed.)

Exit (A) = -0, if unrecoverable error.

WDS - Write sector

Entry driver initialized (SMS called).

(T4) = Channel
(T5) = Equipment
(T6) = Track
(T7) = Sector
(A) = FWA of data buffer. (502 *1 word buffer needed.)

Exit (A) = -0, if unrecoverable error.

All drivers begin at location 600.

```
600         ORG   MSD
600 0000    CON   0       Cleared indicator to cause initial load
```

Use of mass storage drivers is described in detail in Section 7.

*1 For all MS except 808 these are 502 word buffers.

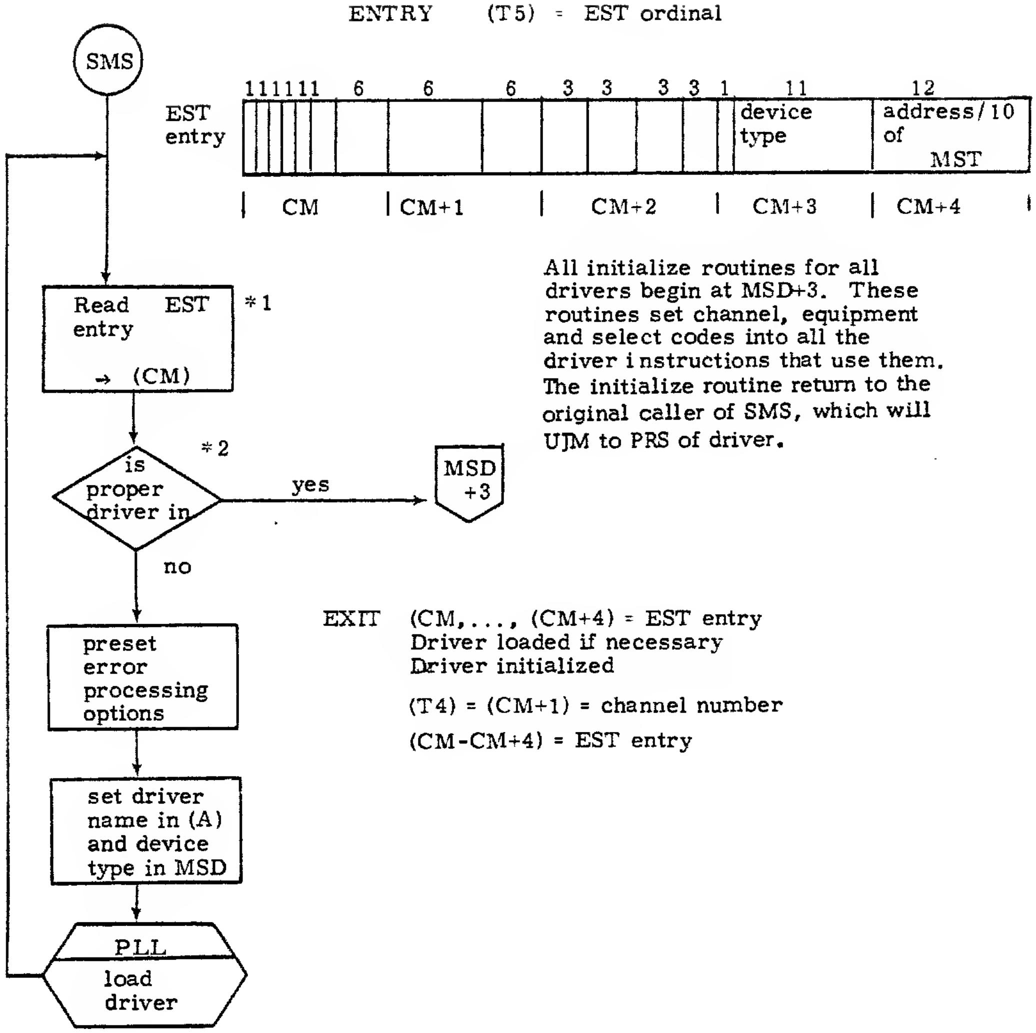
For an 808 it is a 503 word buffer. The extra byte is the expected head position and if it does not agree with actual head position after a Read, we have a Read error.

TABLE 4-3. SYMBOLS USED IN CONJUNCTION WITH MASS STORAGE DRIVERS*

Symbol	Value	Description
MSD	600	MASS STORAGE DRIVER IDENTIFICATION
SLM	601 (MSD+1)	SECTOR LIMITS
POS	606 (MSD+6)	POSITION DISK
WDS	612 (MSD+12)	WRITE SECTOR
RDS	616 (MSD+16)	READ SECTOR
	OTHER MASS STORAGE PROCESSING CONSTANTS	
Symbol	Value	Description
BFMS	7000	SECTOR BUFFER ADDRESS
FSMS	1	FIRST DATA SECTOR OF FILE
	SYSTEM SECTOR ADDRESSES	
Symbol	Value	Description
FNSS	BFMS+2	FNT ENTRY (5 LOCATIONS)
EQSS	BFMS+2+5	EQUIPMENT NUMBER
FTSS	BFMS+2+6	FIRST TRACK
FASS	BFMS+2+11	ADDRESS OF FST ENTRY
DTSS	BFMS+2+12	PACKED TIME/DATE
EISS	BFMS+2+20+5	EOI (END OF INFORMATION) SECTOR NUMBER OF THE LAST TRACK IN THE FILE CHAIN
	MASS STORAGE DRIVER ENTRY POINTS MSD AND SLM ARE READ ONLY CONSTANTS ALL OTHER ENTRY POINTS ARE ENTERED BY A RETURN JUMP	

*Whenever a PPU program desires to read or write mass storage, the program always executes a return jump to SMS. A flowchart of SMS is illustrated in Figure 4-10.

ENTRY (T5) = EST ordinal



All initialize routines for all drivers begin at MSD+3. These routines set channel, equipment and select codes into all the driver instructions that use them. The initialize routine return to the original caller of SMS, which will UJM to PRS of driver.

EXIT (CM,..., (CM+4) = EST entry
Driver loaded if necessary
Driver initialized

(T4) = (CM+1) = channel number

(CM-CM+4) = EST entry

*1 ESTS = FWA of EST

*2 SMS has stored the device type of the present driver in (MSD = 600B) when that driver was loaded. So that it can compare EST device type in (MSD).

Figure 4-10. Set Mass Storage (SMS)

4.10 SPECIAL FUNCTION PROCESSOR (SFP)

Narrative on how SFP is called and what it accomplishes.

In order to understand the SFP autoload this description of the CRM instruction is given here.

CRM

61 d M

read (d) CM words from (A) to M

1. (A) = CM address

   (TO) = (P)

   P = PP address

   Q = (d) = number of CM words to read

   K = TRIP count + op code $\boxed{\text{K is a 9 bit reg}}$

<u>TRIP</u>

0 set up store P in TO and M in P

1 read CM address into pyramid

2 store byte 0 into PP address (P)

3 store byte 1 into (P) + 1

4 store byte 2 into (P) + 2

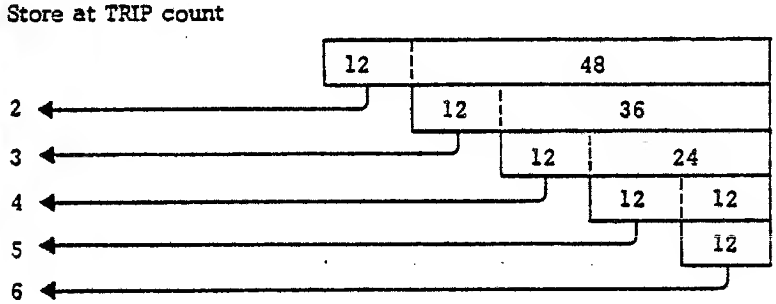
5 store byte 3 into (P) + 3

6 store byte 4 into (P) + 4

7 increment (A), increment P, decrement Q when Q = 0, quit

Read PYRAMID

Store at TRIP count



PLL calls SLT to locate the entry in the PLD which matches the requested PP routine.

	18	6	12	12	12	
PLD+0	name of PP program 1	0	Load Address	Track	Sector	
+1	name of PP program 2	0	Load Address	Track	Sector	
+n	0	RPLA		Length of SFP	LA of SFP	last entry

PLD+n, the last entry is a dummy. SLT reads each PLD entry in (CM, . . . , CM+4).

So, when SLT reads the last PLD entry

CM+0 = 00
+1 = top 6 bits of RPLA normally = 0
+2 = lower 12 bits of RPLA
+3 = CM word length of SFP
+4 = LA of SFP normally = 1073

RPLA is the location in the RPL of the entry for SFP.

SLT discovers that CM = 0 so he exits with (A) = 0

Now, PLL continues with the instruction following the call to SLT.

```
NJN     PLL1
```

Since (A) = 0, this jump is not taken and the next set of instructions will set up the auto load of SFP. (Location 157 in PP Res as of 8/24/73)

LDN	SFPL	set CMR address of SFP auto load code in (A)
CRM	-3,ON	read 1 cm word from CM address SFPL into PP memory starting at PP address 7774B.

The read sets the following locations

This is not direct cell LA but actual load address of SFP hard wired to 1073.

```
7774  6170    CRM    (LA),ON
7775  1073
7776  0100    LJM    (LA)
7777  1073
0000  7773    CON    7773B
```

At the end of the read (A) = SFPL + 1 since CRM increments (A) by the number of words read. During the CRM instruction the (P) are stored at (T0) = location 0000. The read

however forced the constant 7773B in (T0). Now, when the read completes, (T0) is incremented by 1 and stored in (P). So (P) = 7774B. This causes the next instruction to be executed at 7774B. (i.e., an immediate transfer to location 7774B.)

Now the CRM (LA) ON reads 1 word from CM address SFPL + 1 into PP memory at location (LA) = 1073B, at the end of the read the LJM transfers us to location 1073B.

The CRM LA, ON sets the following locations.

```
1073  0000   PSN
1074  20XX   LDC     RPLA which is the FWA in RPL of the routine SFP
1075  XXXX

1076  6113   (CRM)   LA, (CM+3) (CM+3) still set from the read in SLT that started this
1077  YYYY                       sequence, has the CM word length of SFP in RPL.
```

The LDC instruction loads SFP RPL address in (A) and the CRM reads all of SFP into PP memory starting at location (LA) = 1073B. The (P) is set to 1100B (i.e., last instruction at 1076B and 1077B + 1 = 1100B) and SFP starts executing.

SFP is a function processor which is called by the PPR subroutine PLL whenever PLL is unable to find a requested PP program. SFP will check the requested PP package against a table of acceptable PP package calls and, if legal, will call the associated function processor to process the request. The function processors are designed to provide SCOPE 3.4 capability.

Call:

SFP is called directly by the PPR subroutine PLL.

Entry conditions:

(IR - IR+4) = original PP program call.

Functions:

The following are those PP packages who have an associated function processor:

```
STS  -  SCOPE 3.4 STATUS PACKAGE
RPV  -  SCOPE 3.4 REPRIEVE CENTRAL PROGRAM
MSD  -  SCOPE 3.4 SDA/SIS MESSAGE GENERATOR
PFE  -  SCOPE 3.4 EXTEND/ALTER FUNCTION
ACE  -  SCOPE 3.4 ADVANCE CONTROL CARD                 (SFP)
CKP  -  SCOPE 3.4 CHECKPOINT
REQ  -  SCOPE 3.4 REQUEST CALL
DMP  -  DUMP FIELD LENGTH
PRM  -  SCOPE 3.4 PERMISSION CHECKING FUNCTION
D00  -  SCOPE 3.4 ERROR TEXT PROCESSOR
```

NOTE

Functions are discussed in Section 26.

Dayfile messages:

"XXX NOT IN PP LIB." = PP package XXX was not found in PP libraries

"XXX NOT IN PP LIB - CALLED BY YYY." = PP package XXX was not in PP libraries and was called by YYY.

"SFP/XXX PARAMETER ERROR." = Parameter address outside FL.

"SFP.XXX ILLEGAL ORIGIN CODE." = Function illegal for users job origin.

"SFP CALL ERROR." = SFP not loaded by default.

The function processor table format follows:

TFCN - table of function code processors.

ENTRY = 4 WORDS:

	6	6
TFCN+0	PP1 (12)	
+1	PP2	OV
+2	Addr (12)	
+3	bits (12)	

PP1 - 1st two characters of PP program name

PP2 - last character of PP program name

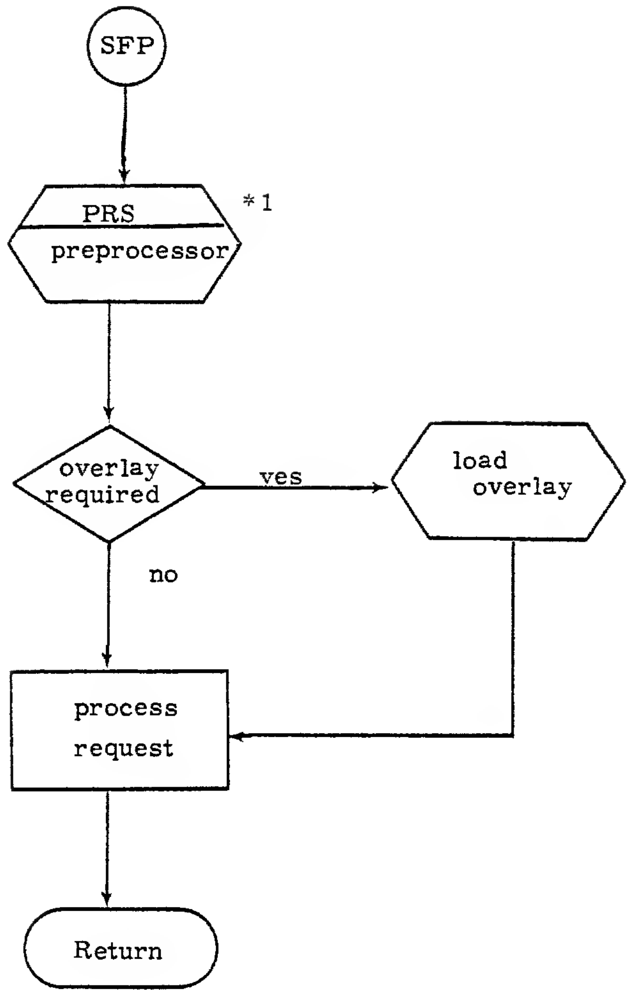
ov - overlay name

addr - address of function processor

bits - bit 11 = check bit. If set, check is performed on bits 0-17 of call for residence within users FL, else bit 0 - 5 = job origin code bits.

Legal Scope Processors

Symbols	Description
STS	STATUS PROCESSOR
MSD	SDA/SIS MESSAGE GENERATOR
RPV	REPRIEVE PROCESSOR
PFE	"ALTER" FUNCTION
ACE	ADVANCE CONTROL CARD
PRM	PERMISSION CHECKING FUNCTION
CKP	SCOPE 3.4 CHECKPOINT REQUEST
REQ	SCOPE 3.4 "REQUEST"
DMP	DUMP FIELD LENGTH REQUEST (URA)
DOO	ERROR TEXT PROCESSOR



*1 SFP may only be called by PLL and not directly for a CP or PP routine.

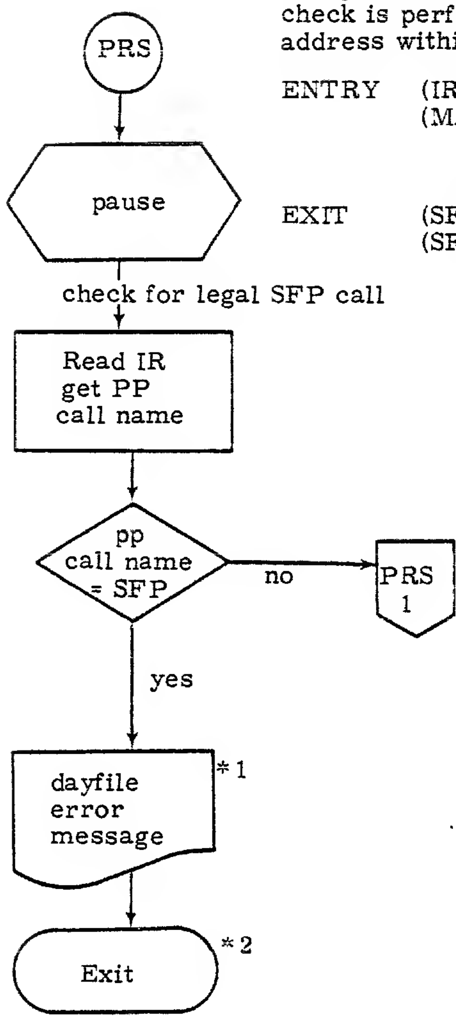
Figure 4-11. SFP Special Function Processor

This is not the PP initialization routine.

Checks the PP package for which "PLL" could not find the associated routine against a table of special functions acceptable. If package acceptable as special functions, a check is performed for valid function code and parameter address within user area.

ENTRY (IR) = address of requesting PP program
(MA) = CM address of PP package name that "PLL" was unable to find.

EXIT (SFPA) = function processor name.
(SFPB) = function processor entry address

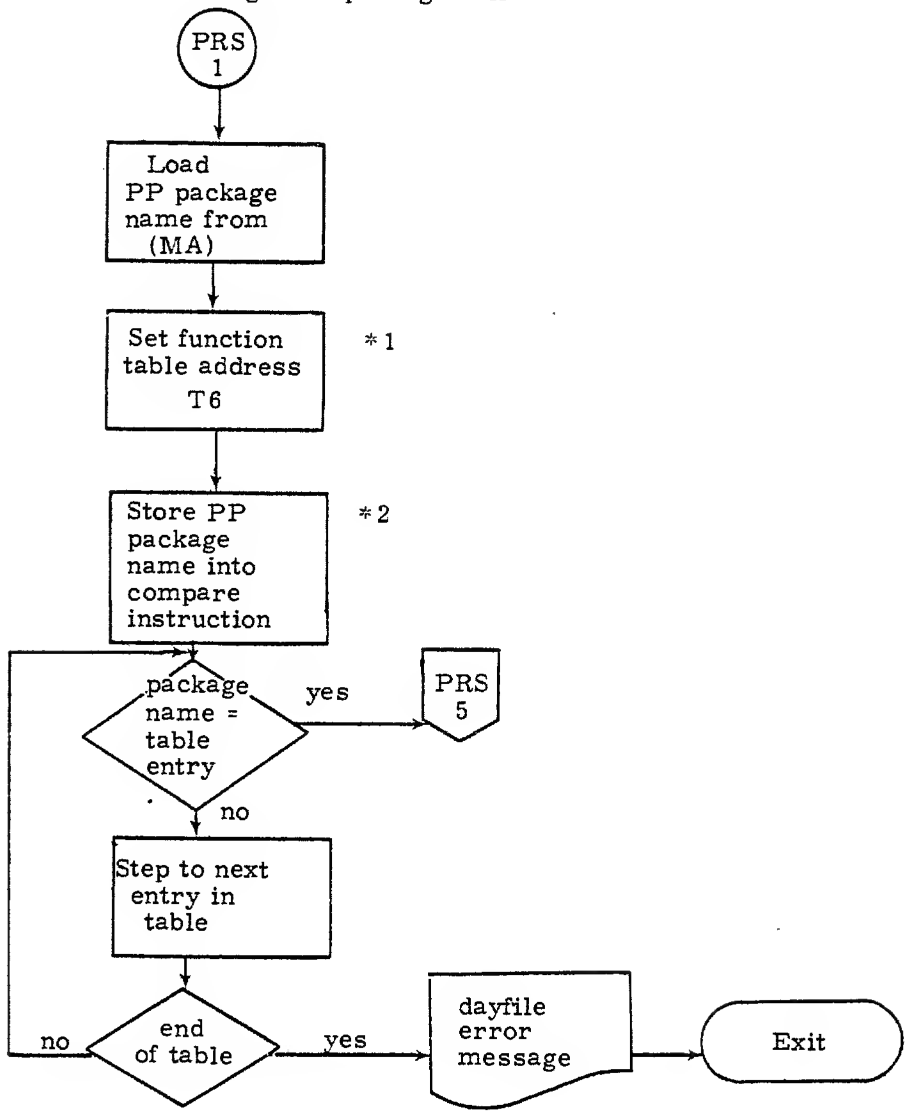


*1 SFP may only be called by PLL and not directly for a CP or PP routine

*2 any error detected by PRS will cause the PP to be dropped and control returned to PPR.

Figure 4-12. Preset Routine (PRS)

check for legal PP package call



*1 uses T6 indirectly to get table entry

*2 compare instruction at PRSA LMC (compare name)

Figure 4-12. Preset Routine (PRS) (Continued)

check origin code for legal operation

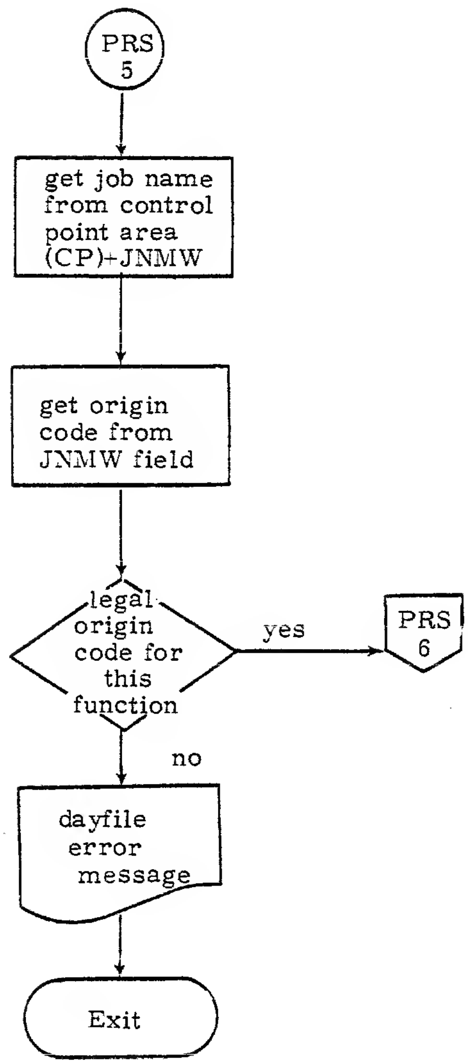
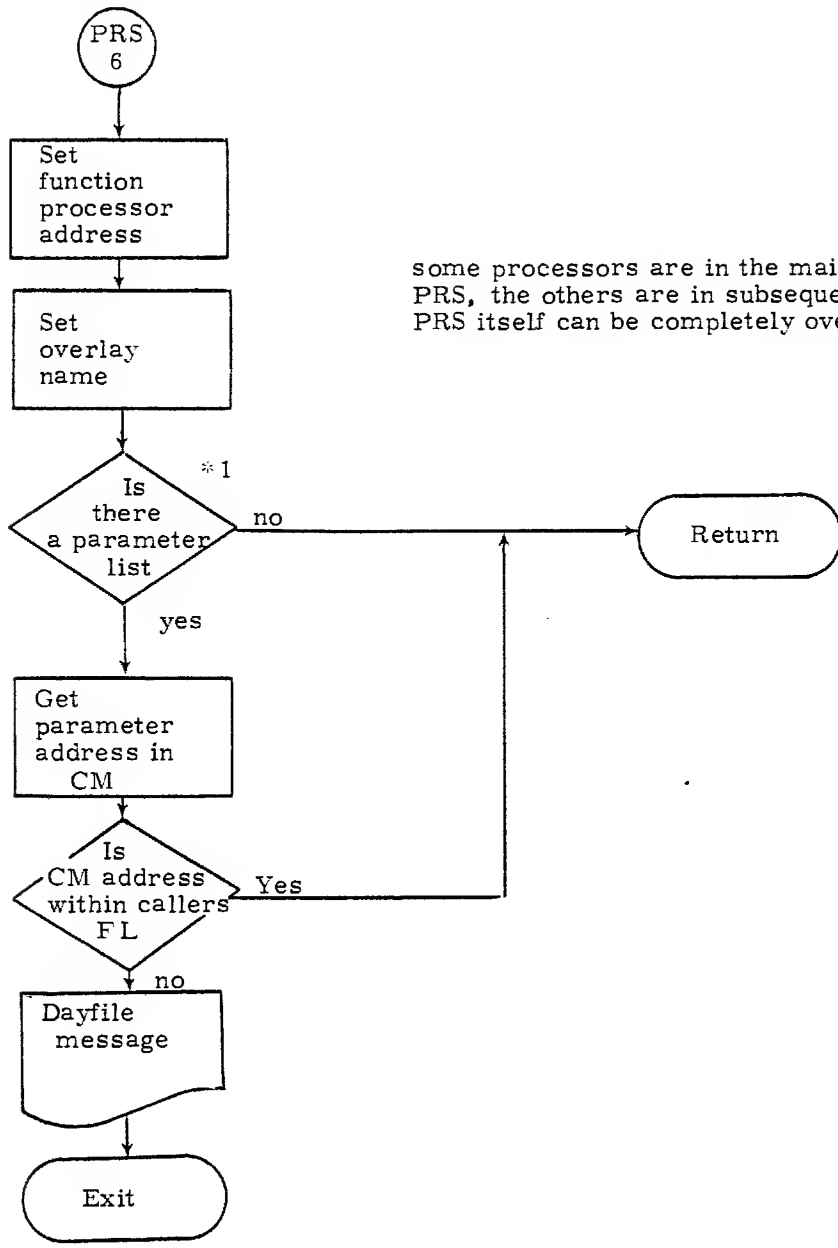


Figure 4-12. Preset Routine (PRS)(Continued)

Legal function check parameters and jump to function



some processors are in the main overlay with PRS, the others are in subsequent overlays. PRS itself can be completely overlayed.

*1 note that this processor was originally a PP request issued from some CPU program.

Figure 4-12. Preset Routine (PRS)(Continued)

JOB PROCESSING 5

5.0 INTRODUCTION

All jobs which flow through the system, whatever their type, will be processed from start to finish by 1SJ, 1AJ, 1CJ, 1RO, 1RI, and (in the case of TXOT) also by 1TA. Flow is controlled by the queue priorities, CPU priorities, in association with time and equipment limits. Depending on the resources desired by the job, all action is initiated, controlled, and eventually error- or end-processed by these five routines.

All jobs are one of the following:

Code	Description
0 = SYOT	System – all jobs entered by the operator at the system console, such as DIS, FST, MY1, STAGE (OPL. T-50), etc.
1 = BCOT	Local batch – jobs entered from all local batch devices.
2 = EIOT	Remote batch – all jobs entered from the remote low speed (EI200 UT) batch terminals.
3 = TXOT	TELEX – all jobs entered via the time-sharing executive program, TELEX.
4 = MTOT	Multi-terminal – all jobs which do one specific task for many terminals while only being scheduled into the system once.

Figure 5-1 illustrates the general system flow for jobs.

5.1 GENERAL JOB FLOW

The priorities are controlled dynamically at the operators console and updated by 1SP (Section 6) which is called by 1SJ. The Job Control area (JBC) in CMR stores the current values of these priorities for the system. Each job can be further restricted by the VALIDUX file, PROFILO file, or job card parameters, but no job can be less restricted than the JBC. 1SP also updates queue priorities in the input and rollout queues, and periodically calls 1CK to checkpoint all MS devices.

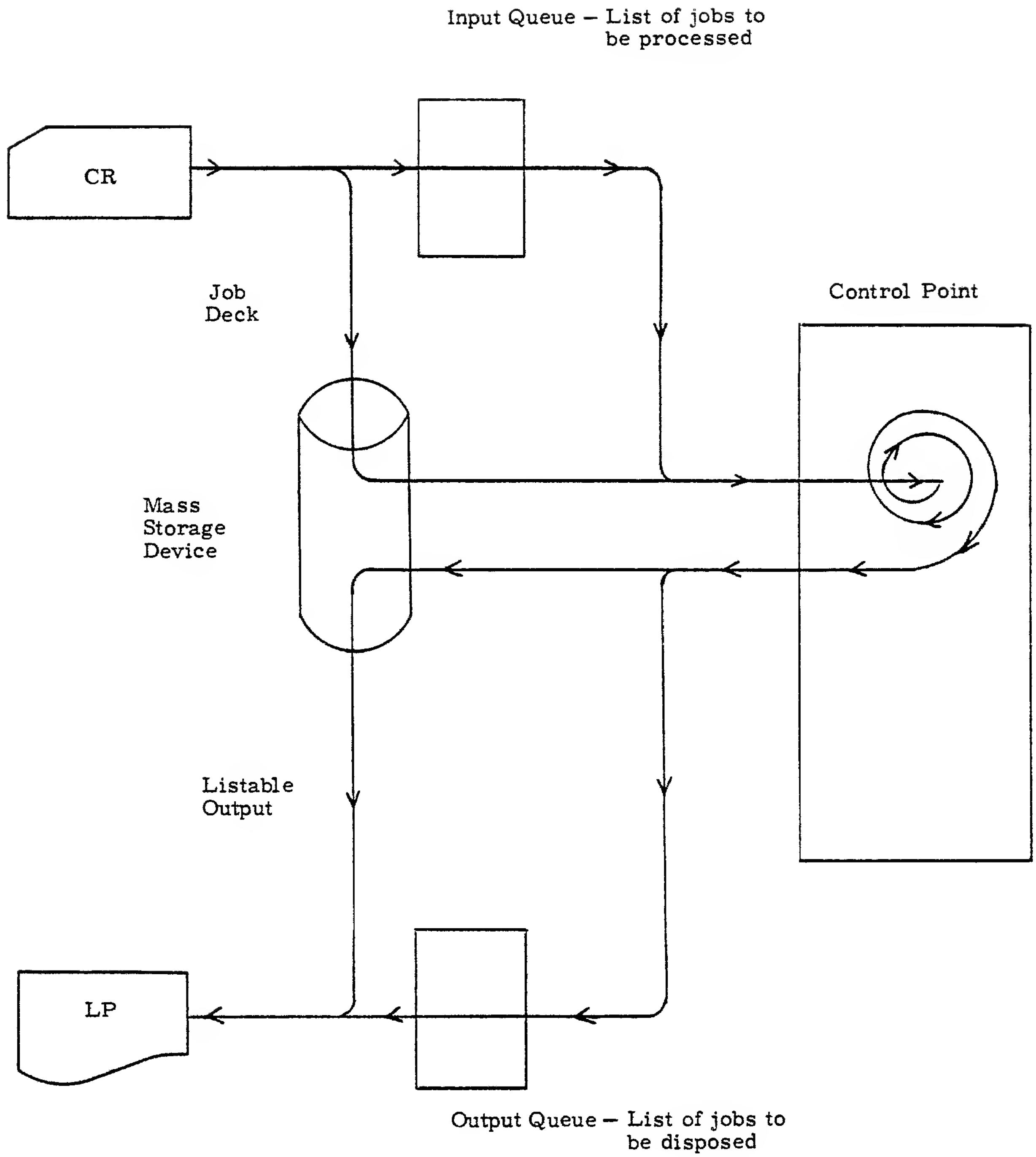
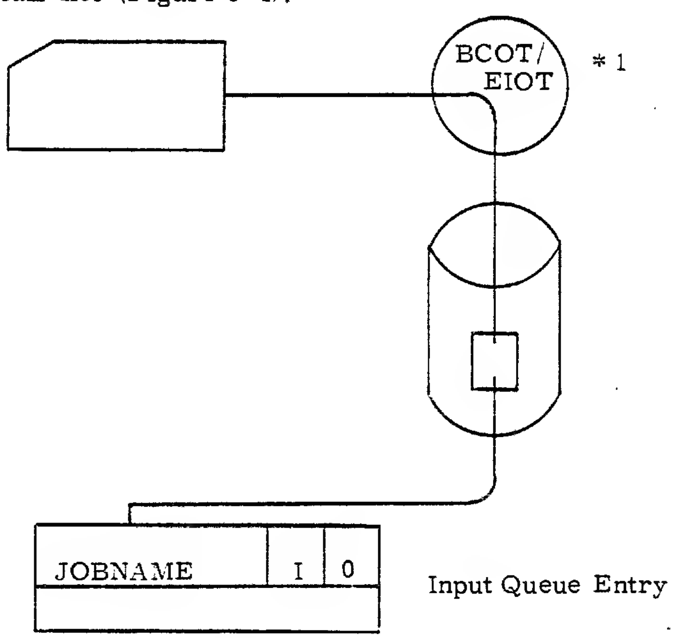


Figure 5-1. General System Flow

Jobs enter the system at the Initial Queue Priority (IQP) for their type (Figure 5-2) As they sit in the input queue, they are aged (i.e., the queue priority is increased until it reaches the Upper Queue Priority (UQP), at which point the priority can be raised no further). At any time, the scheduler, 1SJ, may determine that this job is the best candidate (best job) for a control point by an algorithm that takes into account queue priority and resources desired (FL, etc.), and attempts to schedule or assign it to a control point (Figure 5-3). In order to do this, it will see if there is enough unused core available to satisfy the field length requirements for the job. If not, it will see if there will be enough after scheduled rollouts of other jobs. If not, it will attempt to schedule any other jobs with lower priority than the best job. If there is no way to get enough FL, 1SJ will drop When it is subsequently called again, it may or may not pick the same job as the best job. If there is enough FL, 1SJ looks for an available control point. If there isn't one, he will schedule for rollout any jobs whose priority is lower than the best job. If there are none, 1SJ drops out. When 1SJ assigns the best job to a control point, it will get the FL, set up the control point area (CPA) with information from the VALIDUX, PROFILO, and JBC areas, and will set the input queue priority to (UQP) regardless of what it was when picked. 1SJ will leave the job in no operation status "W", "X", "R" = zero by setting STSW in CPA = 0, and then will call 1AJ (Figure 5-4).



* 1 TXOT/MTOT are started by TELEX and SYOT is initiated by DSD.

Figure 5-2. Read Card Reader

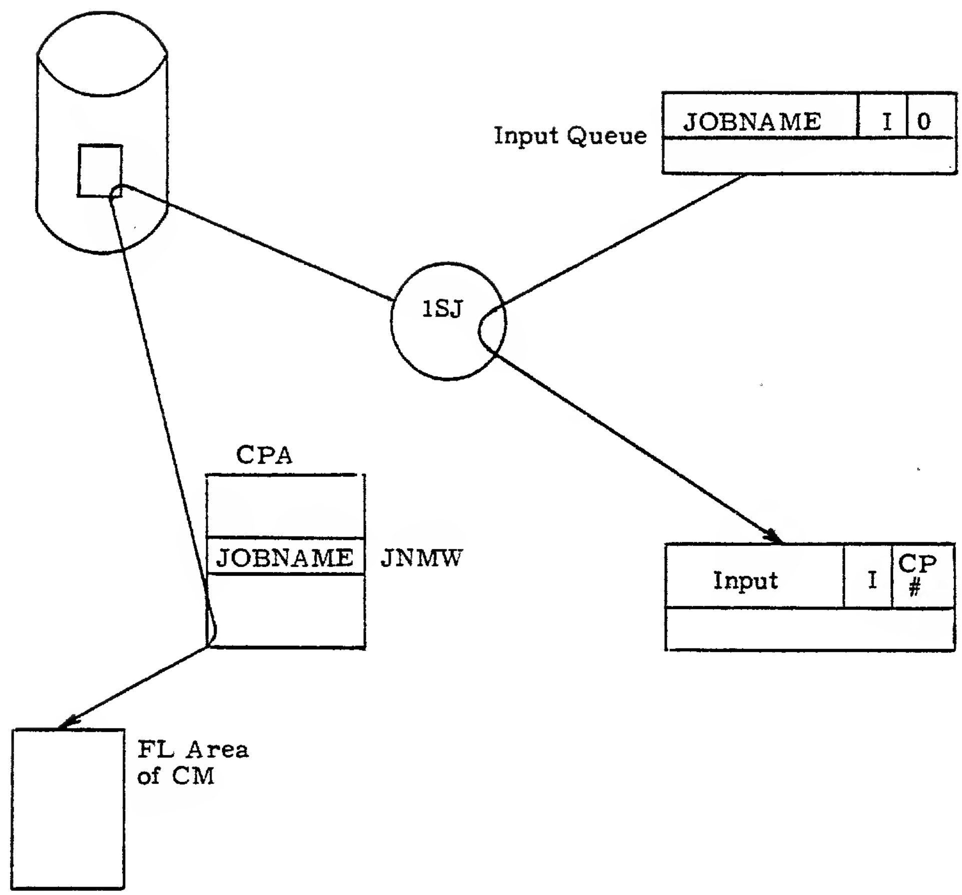


Figure 5-3. 1SJ Prepares a CP for the Job

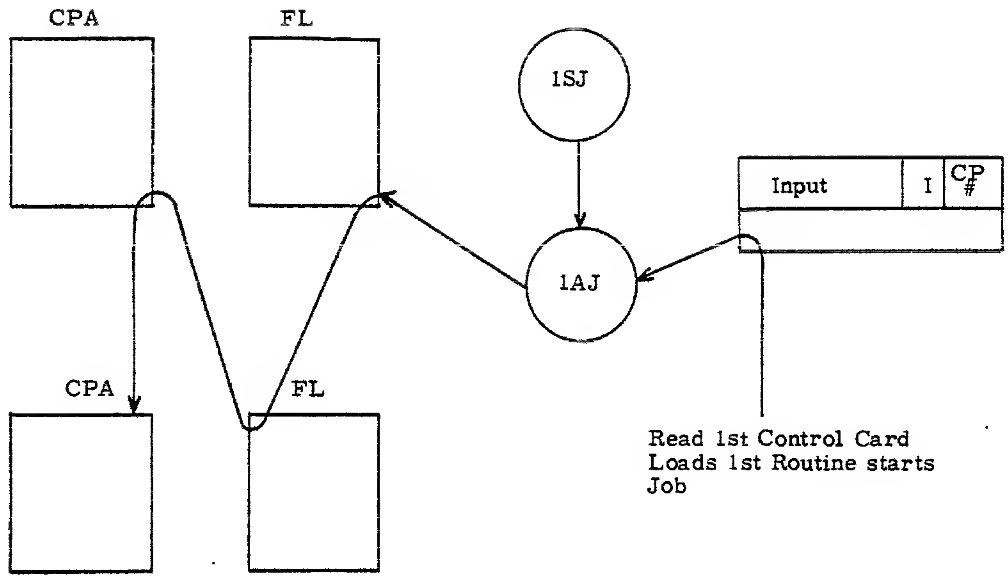


Figure 5-4. 1AJ Starts the Job

1AJ, the job advancement routine, will note that the job status is empty, i.e., last operation complete (in this case first operation is not started). He will call 3AB, an overlay to start this job up. The job can at anytime create local files, and if the name is OUTPUT, PUNCH, PUNCHB, or P8 it will be treated special at job completion time (Figure 5-5).

As the job progresses, CPUMTR and MTR will periodically check all the jobs running at control points and, if either detects "W", "X" and "R" status zero, they will call 1AJ. If the error flag is set, 1AJ will process the error. If the error is not fatal 1AJ will advance to the next control card. If the error is fatal but an EXIT card exists, then 1AJ will advance to the card following EXIT. CPUMTR and MTR also monitor the CPU time-slice, and if the job exceeds its time-slice, its queue priority is dropped to the Lowest Queue Priority (LQP) of that type. This does not mean that the job will lose its control point. If 1SJ finds a best job in the input or rollout queues, then low priority jobs are candidates for rollout. Also 1SJ monitors all the control points, and, if it detects that the CPU time-slice is exceeded before either monitor detects this, 1SJ will lower the queue priority to LQP. An interlock is provided in bit 35 of TSCW in CPA so QP is only dropped once.

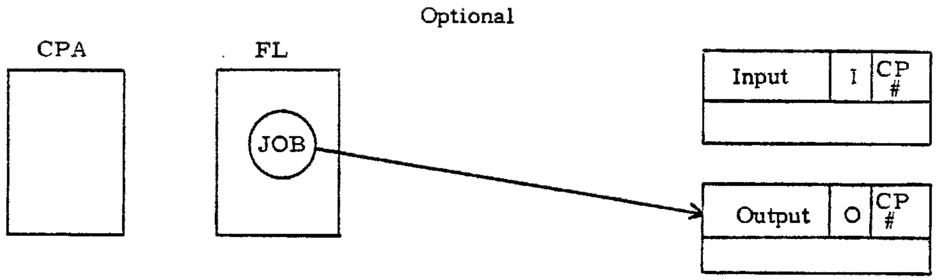


Figure 5-5. Job Creates Local File Name OUTPUT
PUNCH, PUNCHB or P8 - Denoted by OUTPUT type O, however the type will be PRFT or PHFT.

IRO may be called by 1AJ, 1SJ, DIS, Special Entry Point (SEP) calls or some other routines (Figure 5-6). 1RO will dump the job according to the rollout file format (Figure 5-9), will set "W", "X", "R" status to zero, will request the control point be made available, and will release all FL, non-allocatable equipment, i.e., tapes are not released but CP number in est is set to 37B, and all files assigned to this control point. The job is then placed into the rollout queue with whatever queue priority the job had when rolled out. If 1RO is called by an SEP, the rollout file will be called DM* and left assigned to this control point. 1RO will release everything else except the input and control card file, and will call 1AJ to advance the job. In this way FNT space is not wasted while a job is rolled out.

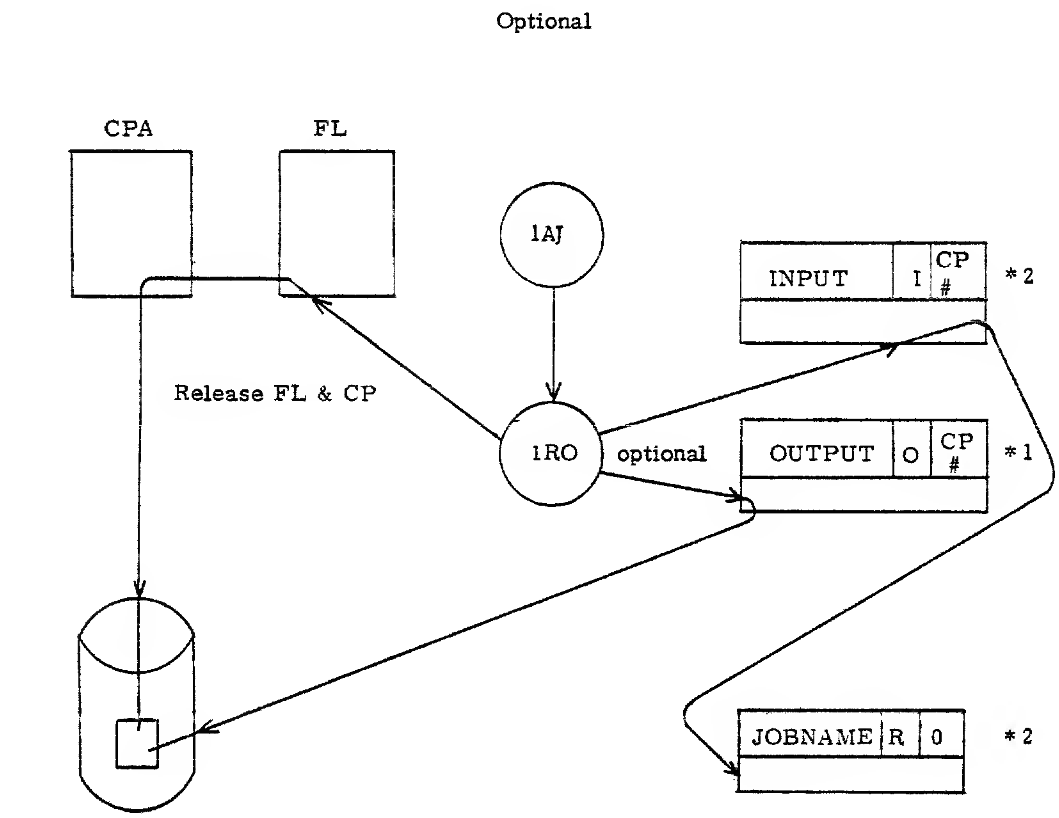
1RI will read the rollout file and re-establish all the files, equipment, etc. to allow the job to continue (Figure 5-7). It will set "W", "X", "R" status to its former values. The control point will now be a candidate for the CPU.

One can say categorically that a job always gets a fresh time slice when it is rolled in. In fact, some problems can occur because of this. If a grinder BATCH origin type job were executing in the CPU, and TELEX origin jobs were constantly bumping it because of higher queue priority, the system always schedules the job back in ASAP since it left with Upper Batch priority with a new time slice. In fact, in a busy system, this BATCH job would never exceed a time slice and would cost system resources by constantly rolling in and out.

When 1AJ detects an end-of-job card stream, a fatal error with no recover, an illegal control card, or some other fatal condition, it calls 1CJ to complete the job. If any of the job flow routines ever detect a type which is not defined (i.e., type not SYOT, BCOT, EIOT, TXOT, or MTOT), it will call 1CJ immediately to end the job. This is protective coding.

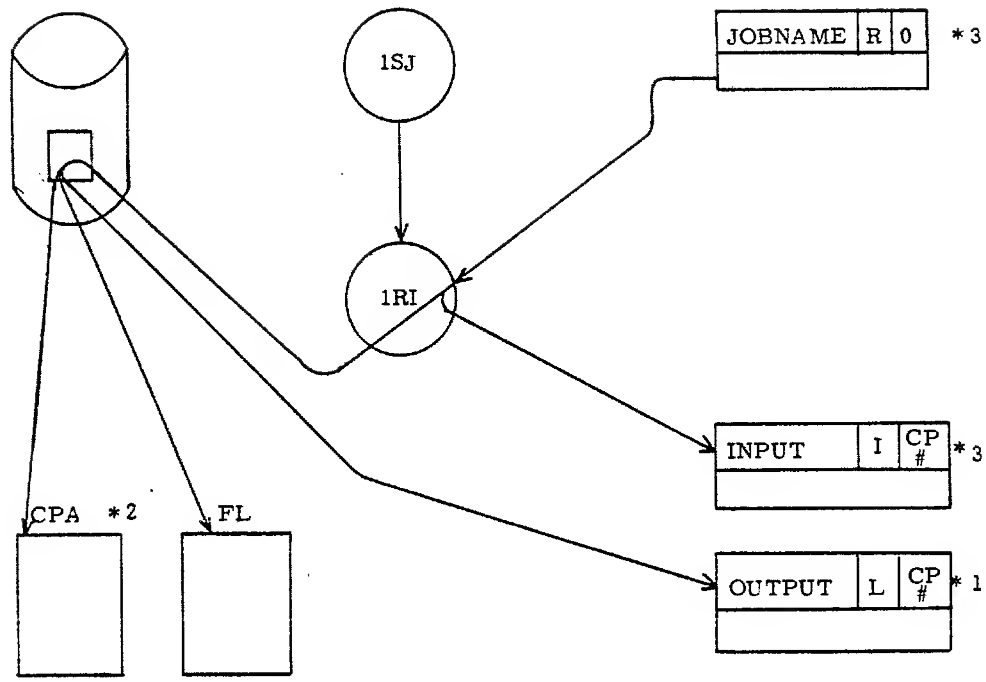
1CJ will locate the local file OUTPUT assigned to this job, if it exists (Figure 5-8). It will then append the job dayfile to the end, write an EOI, and move the file to the output queue, by setting the CP # to zero.

MTR finds a CP in rollout requested status and issues a JACM to call 1AJ.



* 1 And any other Local Files
* 2 This is the same FNT entry

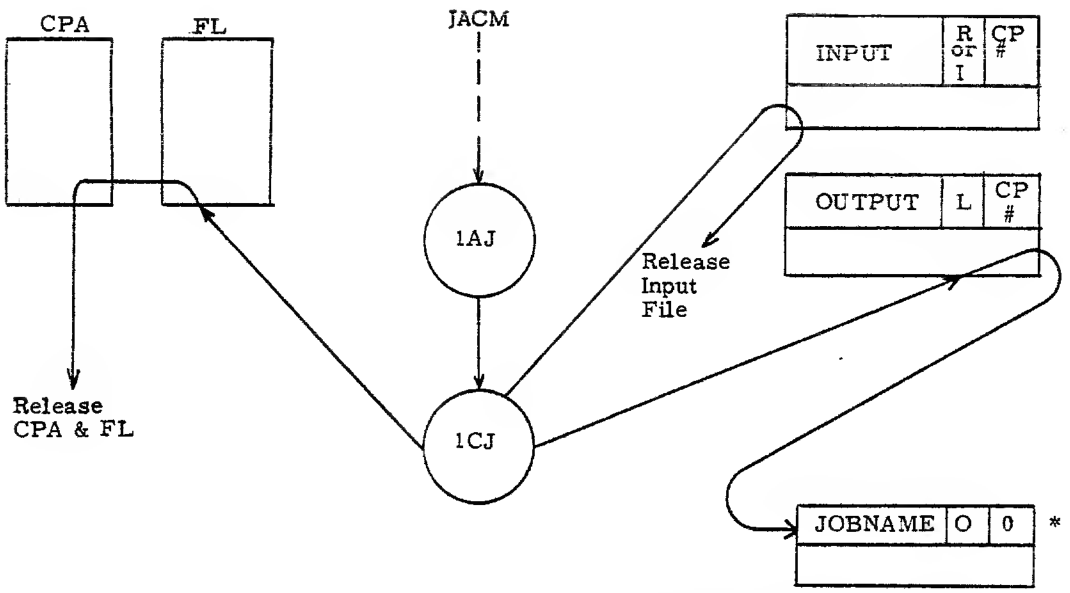
Figure 5-6. Job Is Rolled Out



* 1 And any other Local Files
* 2 Not necessarily same CPA & FL as Figure 5-6
* 3 This is the same FNT/FST entry

Figure 5-7. Job Is Rolled In (From Rollout)

MTR finds a CP in zero status



Change OUTPUT File name to JOBNAME and Type From L to O. Append Dayfile onto end of OUTPUT File.

1CJ Also Returns All Files Associated With this Job Except OUTPUT Type Files.

* 1 Same FNT/FST Entry as Local OUTPUT File

Figure 5-8. Job Completes

5.2 SYSTEM PHILOSOPHY

System philosophy is the description of the concepts relative to rollout file tags, scheduling and sequencing, system sector data location, ageing, and control cards.

5.2.1 Rollout File Tags

Rollout file tags are defined in the common deck, COMSJRO. 1RO, 1RI, and 1TA are the only routines that currently use the common deck.

5.2.2 System Sector

The system sector (Sector 0) for a rollout file contains the following information:

(These symbols are defined in COMSJRO.)

- Dayfile buffer pointer

  The dayfile buffer pointer consists of two words, an exact image of the control point dayfile buffer pointer words from CMR.

Tag	Value	Defined Value	Description
DFBP	50	10*5	Dayfile buffer pointers

- Input file FNT entry

  This consists of a copy of the input file FNT/FST entry. It is zero if no input file is present.

Tag	Value	Defined Value	Description
INFE	62	DFBP+2*5	Input file FNT

- Assigned equipment

  This consists of a list of the equipment assigned to the job. The list is terminated by a zero word.

Tag	Value	Defined Value	Description
AEQE	74	INFE+2*5	Assigned equipment

- Terminal table information (TXOT only)

  The tags for this are TISS and TTSS, where TISS is the terminal table contents at the last rollout, and TTSS is the terminal table contents for the last recovery.

Tag	Value	Defined Value	Description
TISS	240	40*5	Terminal table
TTSS	360	TISS+20*5	Terminal table

5.2.3 Rollout File (Figure 5-9)

The sequence of the rollout file follows:

- Control point area

  The control point section is two sectors in length, and is an exact image of the control point area in central memory.

Tag	Sector Sequence Number	Description
CPAI	1	Control point area

- Dayfile buffer

  The dayfile buffer section is one sector in length, and is an exact image of the dayfile buffer in central memory.

  It is possible to change the dayfile size for the system dayfiles, but the job dayfiles are fixed at 100B words p. II-4-6 in Install Handbook CMRDECK, therefore only 1 sector for this buffer is needed in the rollout file.

Tag	Sector Sequence Number	Description
DFBI	2	Dayfile buffer

- File name table

  The file name table section is n sectors long, terminated by a short sector (logical record). It contains a list of FNT/FST entries of files associated with the control point. The FNT entries are stored as two-word entries in this section.

Tag	Sector Sequence Number	Description
FNTI	3	File name table entries

- Terminal output

  The output for a terminal is n sectors long, terminated by a short sector (logical record). This only exists for TXOT origin jobs.

Tag	Sector Sequence Number	Description
TOPI	4	Terminal output

- Job field length

  The job field length section is n sectors long, terminated by the EOI sector, and is an exact image of the job FL in central memory (n = FL/100B).

Tag	Sector Sequence Number	Description
JFLI	5	Job field
MXFI	6	Job field

TAGS		
DFBP INFE AEQE TISS TTSS	SYSTEM SECTOR . Day file buffer pointers . Input file FNT . Assigned equipment . Terminal table* 1	1 Sector
CPAI	CONTROL POINT AREA 200 CM words	2 Sectors
DFBI	DAYFILE BUFFER 100B CM Words	1 Sector
FNTI	FNT/FST ENTRIES	ℓ Sectors eor * 2
TOPI	TERMINAL OUTPUT * 1	m Sectors eor* 3
JFLI	JOB FIELD LENGTH	n Sectors
MXFI		eoi

* 1 exists only for TELEX origin jobs

* 2 tells 1RI when all FNT entries are accounted for

* 3 tells 1TO when all the terminal output has been issued CIO has a trap for terminal output and will call 1RO instead of sending this output to the output file

Figure 5-9. Rollout File Format

5.2.4 Priority Aging

A job of a particular job origin type waiting in the input, rollout, or output queue is aged if its current priority falls between the lower priority and the upper priority limits.

A job is aged by the scheduler in conjunction with the job control area parameters in CMR. The job control area word illustration follows:

	59	47	35	23	11 0
JBC =	INITIAL QUEUE PRIORITY	LOWER QUEUE PRIORITY	UPPER QUEUE PRIORITY	PRIORITY AGE INTERVAL	CURRENT INTERVAL COUNT

For each cycle of the priority increment routine (1SP), the counter (byte 4 of JBC) is incremented by one. This continues until the counter is ≥ the age increment (byte 3 of JBC). At that time, the job queue priority is aged by one.

All DSD commands, IPRDECK entries and Job Control information is in the KRONOS 2.1 Installation Handbook, Publication Number 60407500A. Specifically, the student should read Sections 5 through 7.

5.2.5 Queues

The Queues (Input, Output, Rollout, etc.) are not separate areas in CMR, but are actually FNT/FST entries in the FNT/FST table area of CMR. When a routine checks a queue, it is actually reading through the FNTs finding those entries that have the type they are seeking but not assigned to a control point.

When a job is moved from the input or rollout queues to a control point, the lfn field of the FNT word contains INPUT instead of JOBNAME. The control point assignment field is set to the control point number and the QP is set accordingly (Upper Input or Rollout priority).

When a job is sent to the rollout queue, the FNT name contains JOBNAME instead of INPUT. The type is Rollout, the control point assignment field is set to zero, and the QP is set to whatever the CPA held at rollout time.

When a job completes, the FNT name (OUTPUT), if one exists, is changed to JOBNAME. The file type is changed from Local to Output, the control point assignment field is set to zero, and the QP is set accordingly (Initial Output type priority).

5.2.6 Rollout Scheduling

When a job is scheduled for rollout, the rollout-request flag, bit 24 in JCIW of the CPA, is set and 1RO may or may not be called. When 1RO is called (by ROCM) it sets the roll-out-in-progress flag, bit 27 in JCIW. When 1RO has rolled the job out, it will reset these bits to zero. Also, if 1RO was called by Special Entry Point routine (SEP), 1RO will set these flags to zero. Also note that an SEP job can be scheduled to be rolled out also. In this case, when 1RO is called, it is a regular rollout, not a response to an SEP.

Many copies of 1RO and 1RI can be run simultaneously, but there is a maximum number to avoid PP saturation.

5.2.7 Scheduler

Only one copy of 1SJ may run at any one time, and this can only be called by the monitor function, RSJM. 1SJ will continue to cycle as long as an RSJM is issued during one of its scans. 1SJ will only cycle some maximum number of times in order to ensure smooth operation.

The philosophy of 1SJ is that any time the status of the system changes, 1SJ should assess the status and modify flow as needed. However, if ten changes occur during one 1SJ cycle, 1SJ is only needed once more (not ten times more) due to the fact that the assessment is made on the total system status, not on just one part. Even if a best job does not get in on one cycle, and may get passed over for the next several cycles due to other higher priority jobs entering the queues after it was picked, it will eventually get to a control point. This is better than queueing up best jobs and saves 1SJ from having another table of priorities to assess, therefore wasting valuable system resources. In a normal mix, eventually all jobs will be scheduled and any minor delay for one particular job will be inconsequential to the throughput of the system in a day.

Figure 5-10 illustrates a typical queue priority scheme. Note that subsystems are greater than MXPS+4 (MXPS=7760). Certain jobs which require that they cannot be rolled out use priority MXPS+1, 2 or 3, such as Permanent File Manager, Local File Manager, etc.

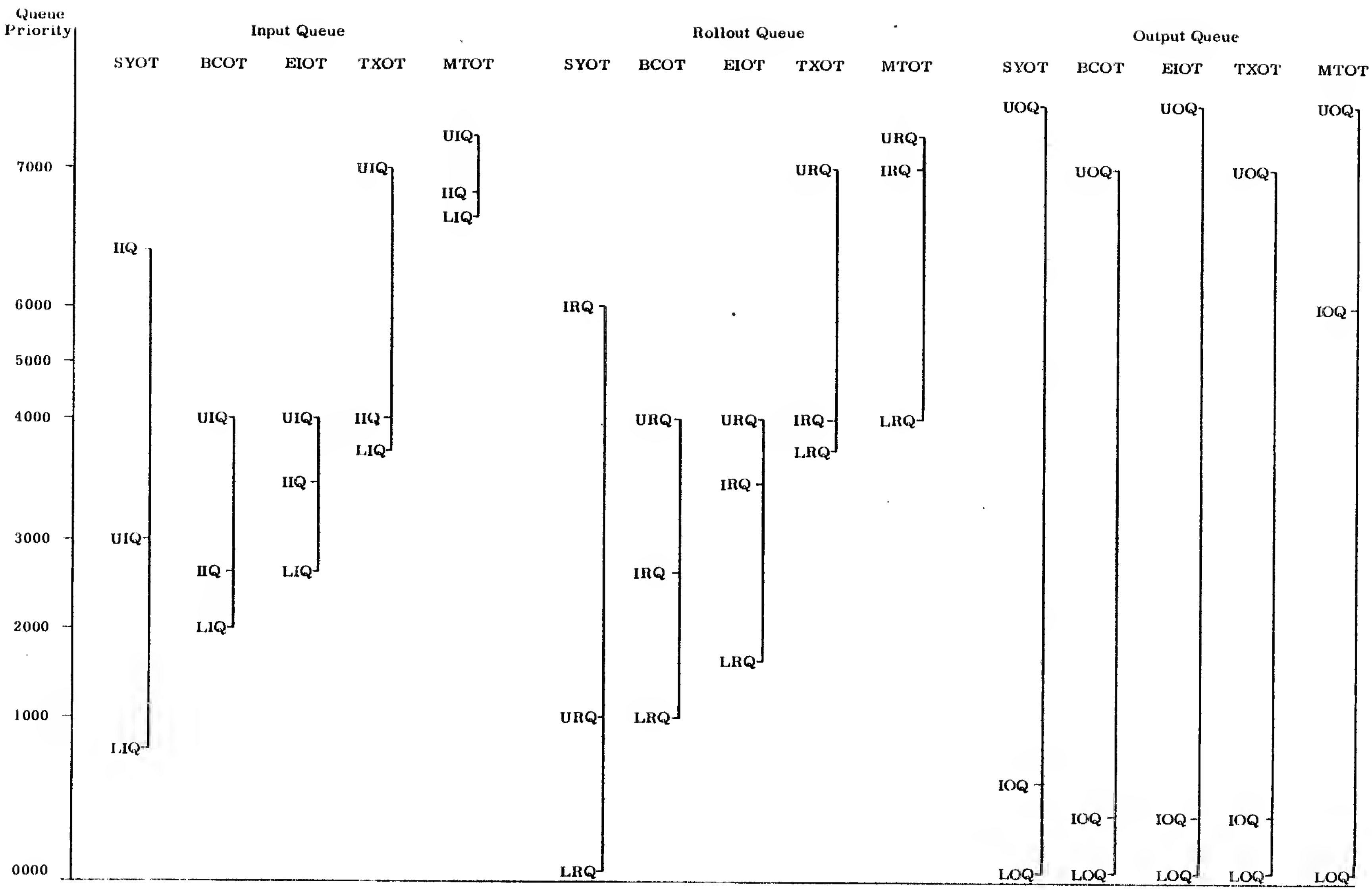


Figure 5-10. Typical Queue Priority Scheme

5.2.8 Control Cards

1AJ has an overlay called TCS which can be called directly from a CP routine or by 1AJ. TCS (Translate Control Statement) will crack a control card and test it for validity. Each control card is actually a call to the system to load a routine whose entry point name is the keyword on the card, such as MODIFY, COPYBR, etc. TCS will assemble the arguments, if any, on the control card and make them available to the routine specified in the keyword. Now, a search is made to locate the routine. First, the FNTs locally assigned to this control point are scanned, then the RCL, then the CLD. If the routine is found in any of these, the first occurrence of the routine is loaded, and the arguments are sent to it, and it is started. (This allows a programmer the facility to define a local program/routine to his control point which may exist in the system already.) If the control card is preceeded by a $ ($MODIFY, $COPYBF, etc.), the local FNT scan is bypassed.

If the entry point name was not found, the RPL and PLD are scanned. If found, the routine is loaded into a PP (set IR = routine name and argument) and TCS goes away.

If no match is found, an appropriate error message is issued to the dayfile and error procedures are initiated by setting the error flags and returning to 1AJ.

Before a CPU program is given control, the loader will place the control card image which called this overlay into cells RA+70 through RA+77. Also, the control card which was cracked by TCS and the parameters found are placed in cells RA+2 through RA+63. If the control card was preceded by a "/", the parameters are cracked in KRONOS format, otherwise they are cracked in SCOPE 3.4 format. All compiler (FTN, RUN, COBOL, etc.) binaries will expect control cards to be cracked in SCOPE 3.4 format.

1) KRONOS format (6-bit ID code)

42	12	6
Parameter (7 characters)	0	ID

ID = 0 for all separators except "=" and "/", and in those cases the character is placed in the 6 bits.

2) SCOPE format (4-bit ID code)

42	14	4
Parameter (7 characters)	0	ID

Parameter = string of characters up to the separator

ID = separator equivalence

0	continuation (for literals)
1	
2	=
3	/
4	(
5	+
6	-
7	Space
10	;
17	Termination) or ...

For example, the control card

MODIFY (I, P=0, N=FILE, A, NR, X, CL)

would be passed as follows: PGNR = RA + 64 = | MODIFY | 11B |

KRONOS 2.1

	42	12	6
RA+2	I	0	
3	P	0	=
4	0	0	
5	N	0	=
6	FILE	0	
7	A	0	
8	NR	0	
9	X	0	
10	CL	0	
11	Binary Zeros		

6-bit code is display character when used and binary zeros when blank.

Full word of zeros terminates control card.

SCOPE 3.4

	42	14	4
RA+2	I	0	1
3	P	0	2
4	0	0	1
5	N	0	2
6	FILE	0	1
7	A	0	1
8	NR	0	1
9	X	0	1
10	CL	0	17
11	Binary Zeros		

4-bit code is binary number.

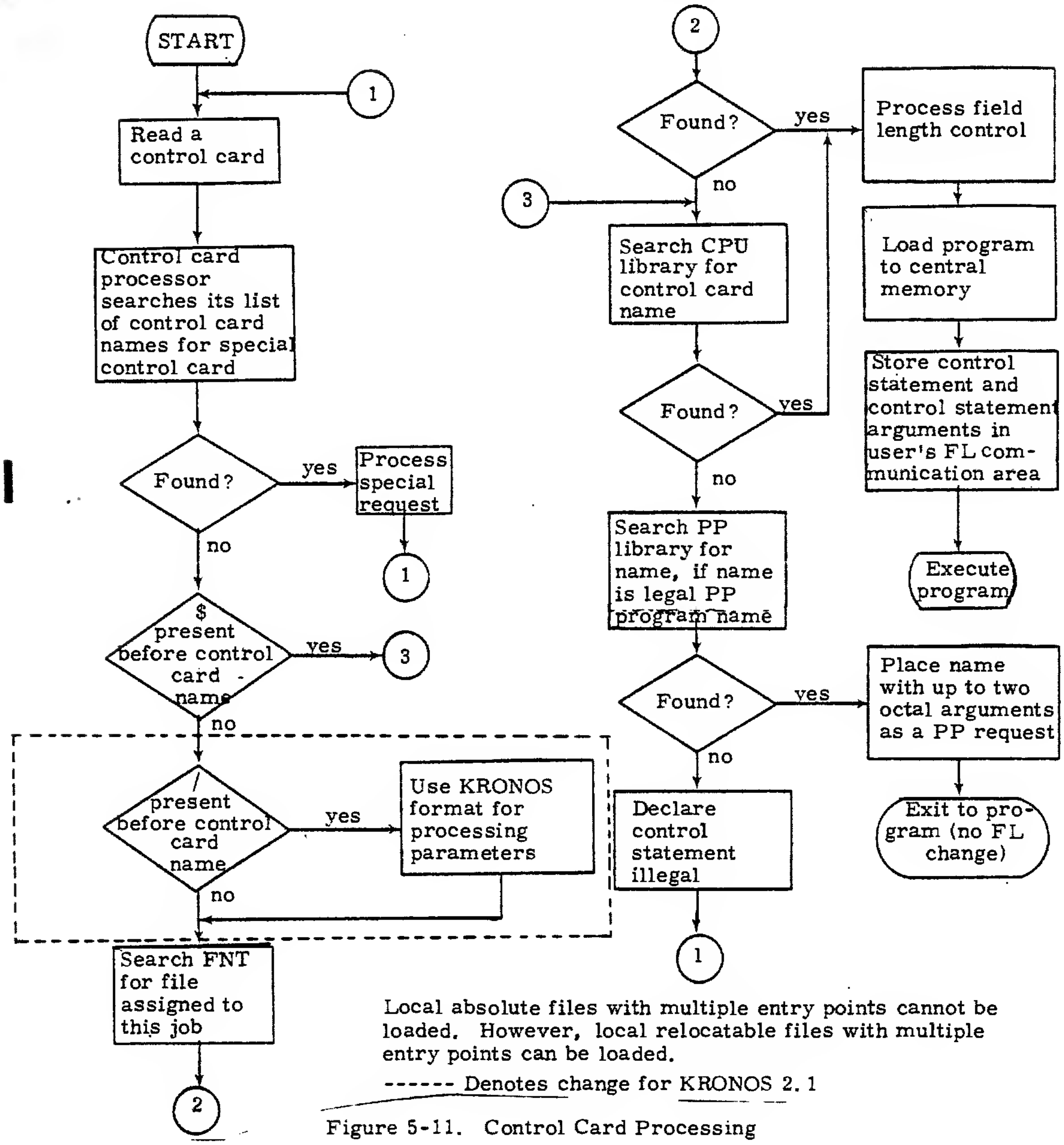
One word of zeros preceded by other than a code 17 implies another control card.

The flow chart (Figure 5-11) shows the flow of control card processing. 1AJ processes CTIME/RTIME directly.

NOTE

Automatic parameter cracking depends on whether the load is from a system or local file.

System Load	Local Load
Default is KRONOS unless *SC specified in LIBDECK.	Default is SCOPE unless "/" on control card.



Local absolute files with multiple entry points cannot be loaded. However, local relocatable files with multiple entry points can be loaded.

------ Denotes change for KRONOS 2.1

Figure 5-11. Control Card Processing

5.2.9 Special file INPUT*.

1. When the user returns the file INPUT, INPUT* is set up to point to the input file, but the user cannot access it.

2. When a CALL (Procedure file) is encountered, the procedure file is expanded on INPUT*.

3. When a procedure file from the system is encountered, a dummy CALL is generated to the CP processor CONTROL and the expanded file is pointed to by INPUT*.

4. When any combination of the above occur, INPUT* is used for all of the above to link up the several "FILES".

   Note: the file INPUT* may not explicitly exist for point 2 or 3, i.e., no FNT/FST entry, but may only be pointed to by CSPW in the CPA. See *5 bit 59 in figure 2-3 word CSPW + 1.

5.2.10 Timed/Event Rollin Processing TEFT.

Overview of TIMED/EVENT Rollin. When a CP program desires to go into timed/event rollout, it uses the ROLLOUT macro and specifies an event and/or a time. 1RO is called to roll the job out and create a TEFT FNT/FST.

<table>
<tr><td colspan="4">Job Name</td><td>Orig</td><td>TEFT</td><td>1</td><td>0</td></tr>
<tr><td>evt des</td><td>eq no</td><td>1st TRACK</td><td>event description</td><td>FL</td><td colspan="3">rollout time period</td></tr>
<tr><td>6</td><td>6</td><td>12</td><td>12</td><td>12</td><td colspan="3">12</td></tr>
</table>

When 1SP is called by 1SJ it will check each entry in the TEFT queue and if the rollout time period has expired it will change the entry to proper ROFT. If time period is not expired, 1SP will use the EATM monitor function to read the EVENT table from MTRs memory. It compares the events with this 18 bit event descriptor and if there is a match 1SP will change the entry to proper ROFT as follows.

<table>
<tr><td colspan="4">Job Name</td><td>Orig</td><td>ROFT</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>eq no</td><td>1st TRACK</td><td>reserved</td><td>FL</td><td colspan="3">Upper queue priority for origin code</td></tr>
<tr><td>6</td><td>6</td><td></td><td></td><td></td><td colspan="3"></td></tr>
</table>

Only PP programs may access the EVENT table via the EATM MTR request. (see p. 4-9 of instant) Therefore, the macro EESET was designed.

EESET MACRO A where A is any 18 bit configuration this macro calls CPM to enter this event description A into the event table. Unfortunately, a job must have SYOT origin to use the EESET macro, see example 2.

The only PPs using this function currently are:

CPM for EESET enter event. See example 2.
0RP to specify when a Write mode PF is not busy (i.e., has no read users using it). See example 1.
0FA to specify when a Write mode FA is not busy (i.e., has no read users attached) See example 1.
1MT to specify when a VSN has been satisfied for RESEX. See example 3.

5.2.10.1 DSD and DIS Commands.

In all DSD file displays the timed/event rollout files will be displayed as TEFT file types. In addition, the Q display has all TEFT rollout files flagged by **.

The DSD command, ROLLIN,XX. may be used to roll-in a TEFT Job.

For a job at control point n, the DSD command n.ROLLOUT,XXXX. will roll the job out for XXXX seconds.

This command to roll a job out for a time period may also be used under DIS as follows:

ROLLOUT,XXXX.

5.2.10.2 Description of Timed/Event Rollout -

The timed/event rollout feature allows jobs to access system resources as they become available. Through use of the ROLLOUT macro, the user may request to be rollout out until an event occurs or time period expires. If the desired event does not occur within the specified time period, the job will be scheduled to roll-in for further processing anyway.

To determine when a specified event has occurred, a system event table is maintained in PP memory. System programs can make entries to this table to indicate occurrence of events. The job scheduler 1SP compares the requested event with the system events recorded in this table to determine if any matches have occurred. If a match occurs, the job scheduler 1SP initiates rollin. If no one is waiting for the system events they are cleared from the table.

5.2.10.3 ROLLOUT - Call format. See Reference Manual.

ROLLOUT - macro follows.

ROLLOUT - macro addr

Entry - If addr is not specified, rollout job until operator initiates rollin. If addr is present, rollout job for time/event described at RA+addr.

	59	29	11 0
addr	0	evd	rtp

Where:

rtp = rollout time period in job scheduler delay intervals ($0 \leq rtp \leq 7777B$). If rtp = 0 the job rolls out for a time determined by the system to insure that the job will roll in if the event he is waiting for is lost or never occurs.

evd = event descriptor.

If evd is non-zero, the event descriptor and rollout time period, rtp, are placed in the control point area (UPCW). When the job rolls out it will wait for the occurrence of the event in evd or the specified time period (rtp) to elapse before becoming eligible for rollin.

If evd = 0, event is taken from control point area and only the rollout time period is taken from RA + addr. This option allows the user to rollout waiting for events that they system specifies.

If evd = 7700XXB, then extended timed rollout is made. (Assume the job scheduler delay is 1 second). Since the maximum time rtp can specify is approximately 1 hour and 8 minutes, the extended time rollout allows the user to roll out for any length of time. This is a strict time rollout with no event dependency. Job rolls out for (777*XX + rtp) seconds.

5.2.10.4 Programming Notes:

It is possible for the central program to get the CPU before the rollout flag is detected by the system. In cases where it is necessary that the calling program know that the rollout has occurred, he should interrogate the UPCW word of the control point area. The lower thirty bits of UPCW = 0 indicate that the job has rolled out and either the event required has occured or the time period has elapsed, i.e., user needs to have RSB capability which is SEP. (See section 5.2.10.6.)

5.2.10.5 Examples:

1. Attempted attach results in file busy.

   a) Assume error processing is set. Upon restarting the job, use of the rollout macro with evd = 0 will rollout for time specified by rtp waiting for the event file ready to be accessed. (ORP enters this event in the system event table when the file becomes not busy. PFM stored the descriptor for this event in the control point area (UPCW) when it found the file busy but it did not set the rollout flag, allowing the user to choose whether to rollout immediately, or to process some other function first.

   b) If error processing is not set the job is automatically rolled out, waiting for the file to be ready to be accessed. When the job rolls back in, the ATTACH request is retried.

EVENT for example 1 is:

6	12
Unit	1st track of file

ATTACH Control Card:

When a user attempts to access files that are interlocked, the system automatically suspends the job until the file becomes available. This is available for the ATTACH function only.

The ATTACH command, and control card, automatically do this and the error flag is not set if the file is busy. The user may bypass this automatic suspension, by specifying the A option on the control card.

If this option is specified ATTACH (lfn/A,....) the system aborts the command if the file is busy. The user calling PFM via the macros provided, can bypass this automatic suspension by specifying error processing. If error processing is specified, the system returns control to the user with error status reflecting FILE BUSY.

2. Job Dependency.

   Suppose that before JOB1 continues processing that he wants JOB2 (a system origin type JOB) to execute a certain function. Assume JOB1 uses the rollout macro with evd = 1300 and rtp = 600. The rollout flag will be set for JOB1 to rollout for 600 seconds or until event 1300 takes place. Before the 600 seconds has elapsed, suppose JOB2 makes the macro call - EESET 1300,* entering the event 1300 in the system event table. JOB1 will then be scheduled for rollin to resume processing. If 600 seconds elapse because event 1300 has not occurred (or the event was cleared from the table before JOB1 rolled out), JOB1 will be scheduled for rollin.

   In any case, JOB1 will not know if it was rolled in because of time or event occurence. Hence, it is necessary for JOB2 to do something, for example write a code word on a PF which JOB1 can check to see if the event occurred.

This job dependency can be accomplished by JOB2 attaching a PF in write mode and then JOB1 doing the same. JOB1 will wait as in example 1 for JOB2 to release the file. However, if JOB2 gets the DPF first it must release the file for JOB2 and then attempt to attach it again. In order to use the EESET effeotively, an installation must change CPM to accept other origin types who issue EESET. This solution has a serious problem, which is possible filling of the event stack. So, a change to CPM warrants careful consideration by the installation to limiting the number of EESET requests per origin type.

3. Tape scheduling with VSN.

   User asks for VSN specified tape. RESEX is called to make the request to MAGNET. Routine CUP in RESEX will issue the ROLLOUT macro and enter timed/event rollout.

   The event is a folded VSN (the sum of the 3 bytes of the VSN truncated to 12 bits, see Level 3 Release KRONOS 2.1 Newsletter p. 5. The rollout times are given on p. 4 of the above document). When the assignment is made, 1MT will enter the event with the EATM function. 1SP will then schedule the job for rollin.

   Event for example 3 is:

6	12
76	Folded VSN or 0 (for-commitment or at initialization to indicate MAGNET is ready.)

   See chapter 9 for discussion of RESEX.

5.2.10.6 The ROLLOUT Macro

The ROLLOUT macro calls CPM function 6 CPM will read the rollout time and event from the users FL and store it into CPA + UPCW. CPM then does a ROCM.

Control is then returned to the user. The user then can execute until the rollout bit is seen by MTR who initiates 1AJ, who calls 1RO. In order to insure the rollout, the user must issue some PP request, since CPUMTR will not honor a PP request for a CP scheduled for rollout. CPUMTR will place the CP in "I", auto-rollout status with an outstanding RA + 1 request. The simplest method is to build a dummy FET and issue the RETURN macro. This will issue an RA + 1 request to CIO.

MTR will notice that this CP is in "I" status and is scheduled for rollout, so it will call 1AJ, who calls 1RO.

1RO will roll the job out and then look at CPA + UPCW. If it is zero this is a regular rollout. If it is non zero, then 1RO.

1RO will roll the job out and then look at CPA + UPCW. If it is zero this is a regular rollout. If it is non zero, then 1RO will build a "TEFT" type FNT and place the event and time limit from UPCW into the FST. 1RO then clears UPCW.

When the job rolls in, MTR will find the CP in "I" status, and an RA + 1 request. MTR will call CPUMTR with a zero request and CPUMTR will then honor the RA + 1 request. In the case of the RETURN dummy, CIO will treat it as a NOOP (file does not exist) and complete. Then the CP can continue.

However, it does not know if it got rolled in for the time period or the event, so it is the users responsibility to insure the event occurs. See RESEX for an example of this above activity.

5.3 SPECIAL PROCESSING

Special Processing is the processing of subsystems, special entry point jobs, and special RA+1 requests.

5.3.1 Subsystems (SS)

An SS is a special type of job with many privileges not granted to regular jobs within the system. Some of the privileges are:

1) SS cannot be rolled out

2) SS can make use of the inter CP communication special RA+1 requests (SIC and RSB) and receiving and sending data buffers.

3) SS can get a CPU priority above normal CP jobs

4) SS need not be restricted by JBC or VALIDUX, however an SS must have a user index set in UIDW, in order to access permanent files.

5) SS determines which CP to run at and storage moves any other job occupying that CP at SS load time.

6) The SS has an implicit special entry point (SSJ=) status

7) SS can request the CPUMTR to load a PP routine whose name begins with a numeric (RA+1 call TLX). (Any PP request from a normal job must be for a PP routine whose name begins with a letter. Any other PP call aborts the CP program).

In order for a job to qualify as an SS, it must:

1) Have a Queue Priority (QP) greater than 7763 and be defined in SYSTEXT SSCL or SSCL+1 in CMR.

2) Have an entry defined in 1DS so that it can be called from a DSD command.

3) Have a unique QP, since it interacts with the system based on its QP and not on its user index or name or control point number.

Job advancement, scheduling, and detecting an SS are different than for normal jobs. 1RO normally cannot roll out any job with QP ≥ MXPS, therefore 1RO is the only job flow routine which does not need a trap for SS. 1SJ must trap the SS in order to ensure it a CP immediately and to assign it the QP. Since an SS normally has a very high CPU priority, an SS programmer must be careful not to be a hog and keep other users out of the CPU.

Since interaction of SS is very intimately associated with the system, and, in many cases, "hard wired in" , it is not very feasible to define a new one; however, since the system was designed to handle 9 SSs, it is feasible to replace one with an SS of your own choosing. As an example, replace CYBERLINK with MYJOB.

1) In SYSTEXT at Correction Identifier (CI), replace PPCOM.53 with MYPS EQU 7765 MYPS.

2) In 1DS for function 33 at (CI), replace 1DS.1217 with VFD 12/MYPS, 18/0L1MY, 6/75B, 12/0
where: 12/MYPS is QP from SYSTEXT. 18/0L1MY defines a start up PP routine. 6/75 defines a relative CP number where 1 = CP1, 2 = CP2, 3 = CP3, and storage moves a job at that CP if necessary. 77 = the last CP, 76 = the next to last CP, and 75 = the second from last CP. However, if those CPs are occupied, one must take a CP that is even lower. 12/0 is a mask used by subroutine SSS in 1SJ to determine if a disable/enable bit is set in SSTL byte 2 in CMR. If mask = 0, then SS cannot be disabled.

3) Write a PP routine called 1MY which must set up the control point area with queue and CPU priorities and other essential cells set. Then request a call to 1AJ. If inter-CP communication is desired, 1MY will need to set up ICAW in the CPA, defining the transfer in and transfer out buffers.

   It can either set up a control card in the control card buffer area or put it into DSD's control card buffer area, and request 1AJ with function 0, process DIS buffer call. The control card should call the CPU overlay which will run as the SS. (The user should refer to 1TD routine printout for an example of these set up calls, CI 1TD 5812 through 1TD 5830, and 1TD 6023 through 1TD 6032.

4) Set up the DSD type-in entry for this SS in the DSD SYNTAX TABLES 9AX, 9AY, 9AZ, 9AO, or 9AW to recognize the type-in, MYJOB. (Refer to Section 25).

5) Modify IPR, the IPRDECK analyser, to recognize that this SS can be autoloaded, if desired. (Refer to Section 24).

6) Set up the subsystem control word in byte 4 of SSCL+1 in CMR. Note that SSCL is allocated in 12-bit bytes which are allocated in decreasing QP order. (i.e., CYBERLINK is last, TELEX is second since the first byte is reserved, EI200 is third, etc.)

SSCL	Reserved *2	TELEX	E/I 200	BATCH I/O	MAGNET	*1
SSCL+1	TRANEX	TELEX Stimulator	TRANEX Stimulator	RESERVED *3	CYBER LINK	

*1 If SS is not active then SS byte = 0, if active then byte = CP number.

*2 Inviolate byte

*3 Used for transient SS such as CMS (initialize MS)

When all of the above is done correctly, the DSD type-in, MYJOB, will:

1) Cause DSD to call 1DS which will

2) Find MYJOB in its table and build an FNT/FST entry with QP=MYPS of type INFT in the FNT tables. This effectively puts my job into the input queue.

	18	6	18	6	6	6
FNT	Controlling PP Routine 1 M Y	CP number Desired 75B	Job Sequence Number	Job Origin =SYOT	"INFT"	*1 0

	6	6	12	12	12	12
FST	id =0	eq*2 =0	First track*2 =0	First sector*2=0	FL Required	QP 7765B

*1 1SJ will set CP assignment in here

*2 These fields refer to job card stream and since SS doesn't have one at this time they are zero.

3) Eventually 1SJ will be initiated and, assuming no other SS jobs of higher QP are in the input queue, will pick MYJOB as the best job. 1SJ will trap this as an SS job in subroutine SFJ (search for job) since its QP is greater than 7763, and will jump to the subroutine SSS in overlay 3SA. SSS will read the FNT/FST entry and the SS control words SSCL and SSCL+1. If the SS byte is nonzero (i.e., SS already active) SSS will clear the FNT and drop 1SJ. If the SS byte is zero (i.e., SS not active), SSS will request the control point specified from CPUMTR. If the control point is not specified, SSS will drop 1SJ. (This is illegal – a CP must be requested). If the control point requested is not available, then SSS will request rollout of the offending job and drop 1SJ. When 1SJ is called again, it will find this SS, transfer to SSS, and find the control point available. SSS will assign this SS to the control point and stuff the control point number in the last byte of the FNT. Protective code prevents an SS from requesting a control point which is not defined in the system, such as 0 or > last control point number. It will then build the CPA. STSW and JNMW are set: JCIW, TSCW and CTLW are set to the maximum; and APUW, ACUW and AACW are set to all 7s (i.e., maximum). It sets CSPW to all zeros, except EOR flag is set (i.e , no input file). It clears all the rest of the CPA. This effectively gives the control point unlimited access. It then

clears the 1SJ active flag, JSCL+1, and requests 1SJ with monitor function RSJM. It will put the requested start-up routine, 1MY, into its IR and exit to PPR (i.e., start-up 1MY in this PP) 1MY must reside in RPL or PLD.

4) 1MY will set up the CPA whatever way it wants it. It then sets up a control card call to MYJOB, the controlling CPU routine, which must reside in the RCL or CLD (via SYSEDIT or from the dead-start tape). Then 1MY will exit to 1AJ with the parameters set to specify where the control card(s) is located. 1MY may either drop or stay depending on how the SS will use it or other PP routines. (It may also request required FL.)

5) 1AJ will advance the job by loading the CPU overlay requested (MYJOB) absolute binary, and starting up the CPU. After the loader has loaded the routine, control will be passed to it.

6) Now, MYJOB is in control and can run as an SS privileged control point. It is a good idea to have a control card stream with an EXIT card followed by a control card call to some recovery routine, in case the SS sets an error flag. Otherwise, the SS will be aborted on a fatal error.

5.3.2 Special Entry Points (SEP)

Many functions normally performed in a PPU can be done better in the CPU (such as DMP). However, a normal CPU program is too restricted to perform these functions; hence, the concept of SEP is provided.

An SEP program runs in the CPU and is able to:

1) access privilege files (VALIDUX, PROFILO, etc.)

2) access CM outside of its normal FL via (SIC and RSB requests. See para. 5.3.6 for the format of these calls).

Anyone can code an SEP routine, but it must be SYSEDITED onto the system in order to work. A local file with any SEP processors would be treated as a normal entry point, however when they attempted to do a privileged SEP procedure, they would be aborted.

The procedure for writing an SEP routine is as follows:

1) CP code the routine using one of theSEPs defined later.

2) Write the program in ABS format. (Only ABS type binaries may contain SEPs.)

3) SYSEDIT the job onto the system.

4) Run the job by calling it via:

   a) A control card request,

   b) an X. SEP. console command (e.g., n.DMP, etc.), or

   c) an RA+1 request.

This method only works in the following two ways:

1) The RA+1 request is for a processor that is in the SFP category or modified to be in the SFP category. (See overlay 2SG, SRP – Special Request Processor in SFP.)

2) The RA+1 request is for a user-supplied PP routine which will get the absolute deck from the system, set SPCW, and call 1AJ to process a function 3 request (1AJ call from other PP routines).

SYSEDIT processes the SEP in the following manner:

1) The binaries are loaded onto the system device (either during deadstart or appended onto the running system file) and a normal CLD entry is made.

2) One extra word is appended onto the end of the CLD entry. Its format is:

	1	5	1	1	1	1	1		18	13
SEPA =	A	0	B	C	D	E	F	0	DA	SA

Where:

A Set to indicate special entry point table entry.

B Set if ARG = entry point present.

C Set if DMP= entry point present. ("DA" is associated parameter.)

D Set if SDM= entry point present.

E Set if SSJ= entry point present. ("SA" is associated address.)

F Set if VAL= entry point present.

DA = VFD 2/0, 1/S, 1/C, 1/F, 1/U, 12/FL

Where:
- S = Suppress DMP= on control card call.
- C = Only create DM* with nothing on it.
- F = Dump FNT entries, CPA and field length, to file DM*.
- U = Create file DM* as an unlocked file.
- FL = 0, for dump of full FL. FL, for dump of FL* 100B of FL.

All normal ABS entry point names in the CLD will have bit A = 0.

The following system functions are SEPs:

1) CHECKPOINT/RESTART
2) DMP – dump field length
3) SCOPE 3.4 PP requests
4) RESEX – resource allocation

The following table summarizes these functions:

Special Program Requests

RA+1 Request	PP-Request Processor	CP-Request Processor	
CKP	SFP	CHKPT	(SCOPE products checkpoint)
DMP	SFP	CPMEM	(field length dump)
REQ	SFP	RESEX	(SCOPE request macro call)
LFM/PFM	LFM/PFM	RESEX	(tape/removable pack requests)

The flow of an SEP request is:

1) The RA+1 request must be made with auto-recall

   NOTE: Monitor will force auto-recall on all RA+1 requests, except CIO, unless the job has QP greater than MXPS.

2) SFP processes CKP, DMP and REQ because they are not present in the PP library directory and SFP is called for all requests not found there.

3) Once the PP-request processor (SFP/LFM/PFM) decides it requires the service of a CP-request processor, the PP must set up the following, and then terminate –

   a) Set special processor call word in the control point area:

	18	1	1	1	1	2	12	6	18
SPCW =	Entry point name	A	B	C	D	O	Status return	O	PBA

   Entry point = entry point name of CP processor to be called

   A = Special Program Request active (set only by 1AJ)

   B = RA+1 to be cleared before program reload, if this bit is not set.

   C = Remainder of word is parameter list and not address of same.

   D = If set, do not start CPU back up (set only by 1AJ on control card calls).

   Status return = Will be set by CP processor at completion.

   PBA = Parameter Block Address (must be within FL).

   The PP-request processor must check the upper 12 bits for zero and set SPCW only if it is clear. If not, it indicates SEP already at this CP.

   b) Set rollout flag in control point area (ROCM function).

c) If CP is to be restarted upon completion of CP-request processing, the PP-request processor must set any completion or status bits in CM before terminating. If a PP-request processor is to be recalled upon completion of CP-request processing, the PP-request processor must set the 'B' field of the SPCW word.

d) All PP processors must write their input registers back to RA+1 (using 'B' of SPCW to clear it if desired) in order to get it passed to the CP processor.

4) Once the PP-request processor terminates, 1AJ will find the call word set (SPCW).

   a) The CP-request processor must:

      1) Contain an entry point for each possible call to it via the call word (e.g., RESEX has entry points, SFP, LFM, and PFM).

      2) Not activate another CP-request processor while it is active.

   b) 1AJ calls 1RO to process the DMP= entry point, if one exists.

   c) 1AJ takes the address from the call word and retrieves a 20B word parameter block from that CM address (which can be a FET, parameter list, etc.). Only available if DMP = specified.

   d) The CP-request processor is then loaded.

   e) Once the CP program is loaded, the CM changes shown in Figure 5-12 are made. Note that SEPW = SEPA is set up at this time, p. 5-23.

5) Once the CP-request processor completes its task

   a) It sets any status to be returned in RA+27B(SPPR). This will then be set into the call word (SPCW 24/0, 12/status return, 24/0

   b) It sets the event descriptor, if desired, via EESET macro in COMCMAC. This is done when the CP job must wait on some action to complete before continuing. The timed event queue will roll the job out and in every minute (any selected time) until the action has occurred. (Like an automatic periodic recall cycle). For example, RESEX will set the job in the timed event queue on a VSN command if the required tape is not mounted.

   c) It terminates normally.

6) 1AJ will discover that a CP-request processor has completed and will call 1RI to:

   a) Retrieve status and parameter block from CM (RA+27 - RA+47) PBA only available to DMP= type job.

   b) Reload control point area and field length from DM File, if DMP = exists

   c) Reset updated parameter block back into CM

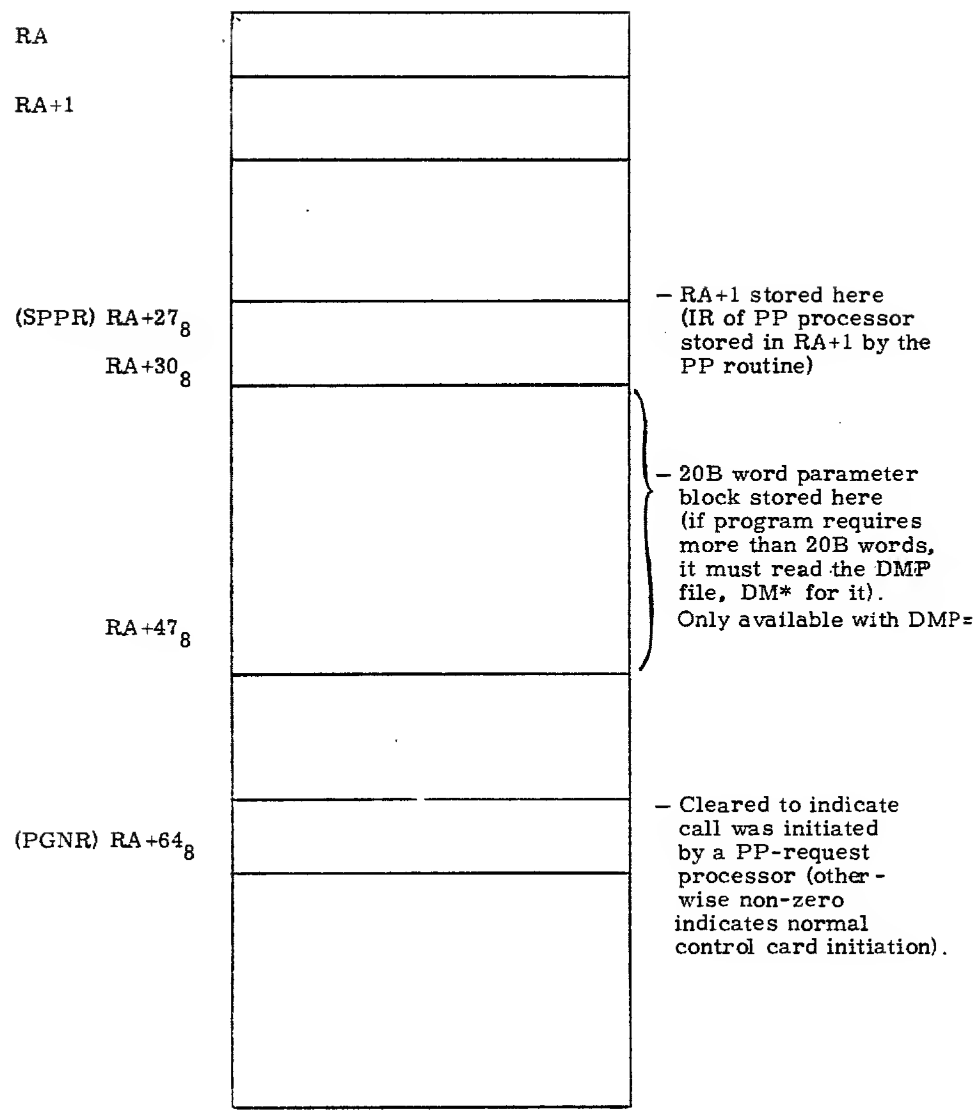


Figure 5-12. Field Length of Loaded CP-Request Processor

d) Update control point area (selective areas)

1) Clear call word

2) Set return status

7) Then 1AJ will advance the job, which may merely restart it where it left off (in the case of the System DMP macro).

8) Any SEP can be rolled out while it is processing. 1AJ 1RO, 1RI will process this as any other job. DM* file and any other file FNTS plus the SEP jobs FL will go to a rollout file, and the job will be rolled out. Normal rollin will proceed when the SEP job is rolled back in.

5.3.3 Special Entry Point Definitions

The following special entry points are available in KRONOS 2.1 for ABS system programs. The routines SYSEDIT and 1AJ process these entry points.

ARG=	Suppress argument
DMP=	Dump previous job before load
RFL=	Automatic field length assignment
MFL=	Automatic minimum field length assignment
SDM=	Suppress control card dayfile message
SSJ=	Define job as special system job
VAL=	Define job as a validation processor

Once a job containing these entry points has been loaded, 1AJ will set up SEPW in the control point area.

	1	5	1	1	1	1	1	13	18	18
SEPW=	X	O	A	B	C	D	E	O	O_{DA}	O_{SA}

Where:

X	–	SEP active flag
A	–	ARG= present
B	–	DMP= present
C	–	SDM= present
D	–	SSJ= present
E	–	VAL= present
SA	–	SSJ= parameter block address
DA	–	DMP= options

NOTE: This is a copy of the SEP entry point word in the CLD.

1) ARG=

Used for a job wishing to do its own control card argument processing. If present, arguments are not passed to RA+2, but the entire control card image, including statement label and other options ($, /), is placed in RA+70.

2) DMP=

A program using this entry point should set up the DA field in SEPW with a PP routine (in the case of the control card or macro DMP it is done automatically) as follows:

	1	1	1	1	1	1	12
DA =	R	0	S	C	F	U	FL

Where:

R — Restart Rollin

S — Suppress DMP= processor if control card call

C — Set, indicates create file DM* only

F — Set, indicates dump FNTs along with control point area and field length. (Complete File)

U — Set, indicates create file DM* in unlocked mode i.e., write mode.

FL = 0, dump entire field length

$\neq$ 0, dump FL* 100_8 of field length (This is a 12-bit field and represents FL/100 desired)

The DA field can also be set at assembly time by using the instruction EQU as follows:

DMP=	EQU XXYYYYB
Where XX =	bit configuration of R0SCFU
and YYYY =	field length specification

The DM* file is the rollout file. The only difference is in the FNT. If it were a rollout file, then the FNT would be:

jobname	origin type	type ROFT	CP = 0 number

However, as a DM* file the FNT would be;

DM*	origin type	type LOFT	This CP number

and the file remains attached to this CP.

DM* is not a legal file name, and a CP user may never create a file whose name contains special characters. However, a CP routine may read or write such a file if it already exists. Hence, 1RO must be asked to create the DM* file if an SEP job will need to use the file. See RESTART in Section 22 for an example of using an empty DM* file created by 1RO.

The parameter list defined previously is only processed on DMP= SEP, and is actually moved to RA+SPPR+1 by 1RO.

The flow of a DMP= request is:

a) 1AJ will find this control point idle.
   "W" = "X" = "R" = 0 or DIS will call 1AJ directly.

b) 1AJ calls 1RO, which creates a rollout file as specified in the DA field of SEPW. The file will be named DM* and left attached to the control point as a local file.

c) 1AJ then loads the CP program specified in SPCW.

d) DMP will dump the field length and CPA as requested in DA from the DM* file. When DMP is complete, the "W" = "X" = "R" status in the CPA will go clear.

e) 1AJ will be called to advance the job; it will see that a DMP= has just completed and call 1RI to restore the control point FL and CPA from the DM* file.

f) 1AJ will advance the job or restart the previous job.

Figures 5-13 through 5-15 show a graphic picture of the procedure while Figures 5-16 through 5-22 illustrate the flow charts for this procedure.

3) <u>RFL=</u>

When a program with RFL= is loaded from the system, the program's field length is set to the value of RFL= (rounded to the next higher 100_8).

4) <u>MFL=</u> (Minimum FL)

Same as RFL= except nothing is changed if the existing field length is greater than the MFL = value. (i.e., if present FL>"MFL=", then no FL change.

5) <u>SDM=</u>

For programs with SDM= entry points, no dayfile message is generated on the control card call. The program should issue its own messages. Using ACCFAM as an example, the password on an ACCOUNT card should not appear in the dayfile. When ACCOUNT, ABCUSER, PASSWRD. is issued, ACCFAM using SDM= entry point can strip off the password and issue ACCOUNT,ABCUSER,. to the dayfile.

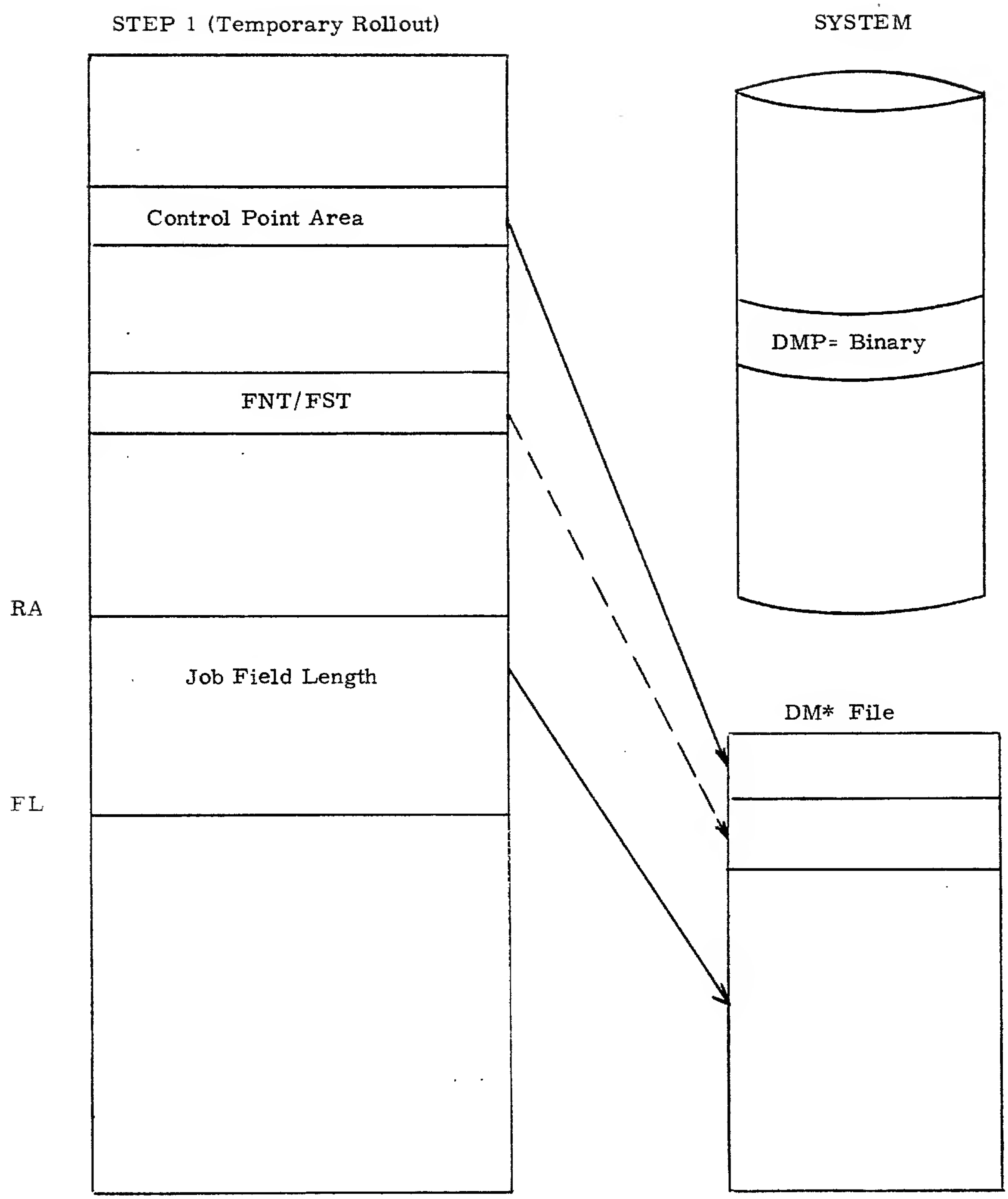


Figure 5-13. DMP= Processing (1AJ Calls 1RO)

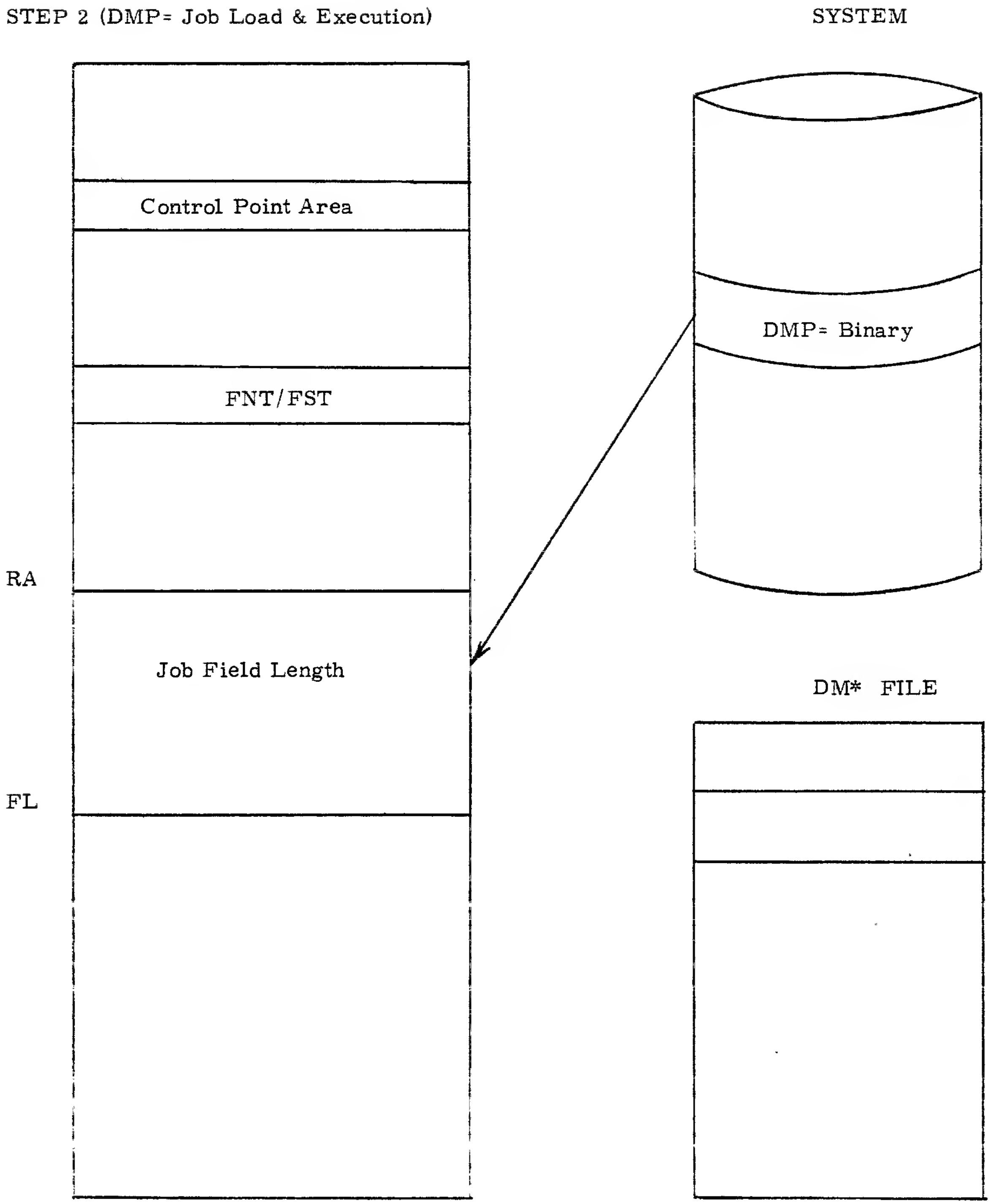


Figure 5-14. (1AJ Call LDR to Load DMP= Program)

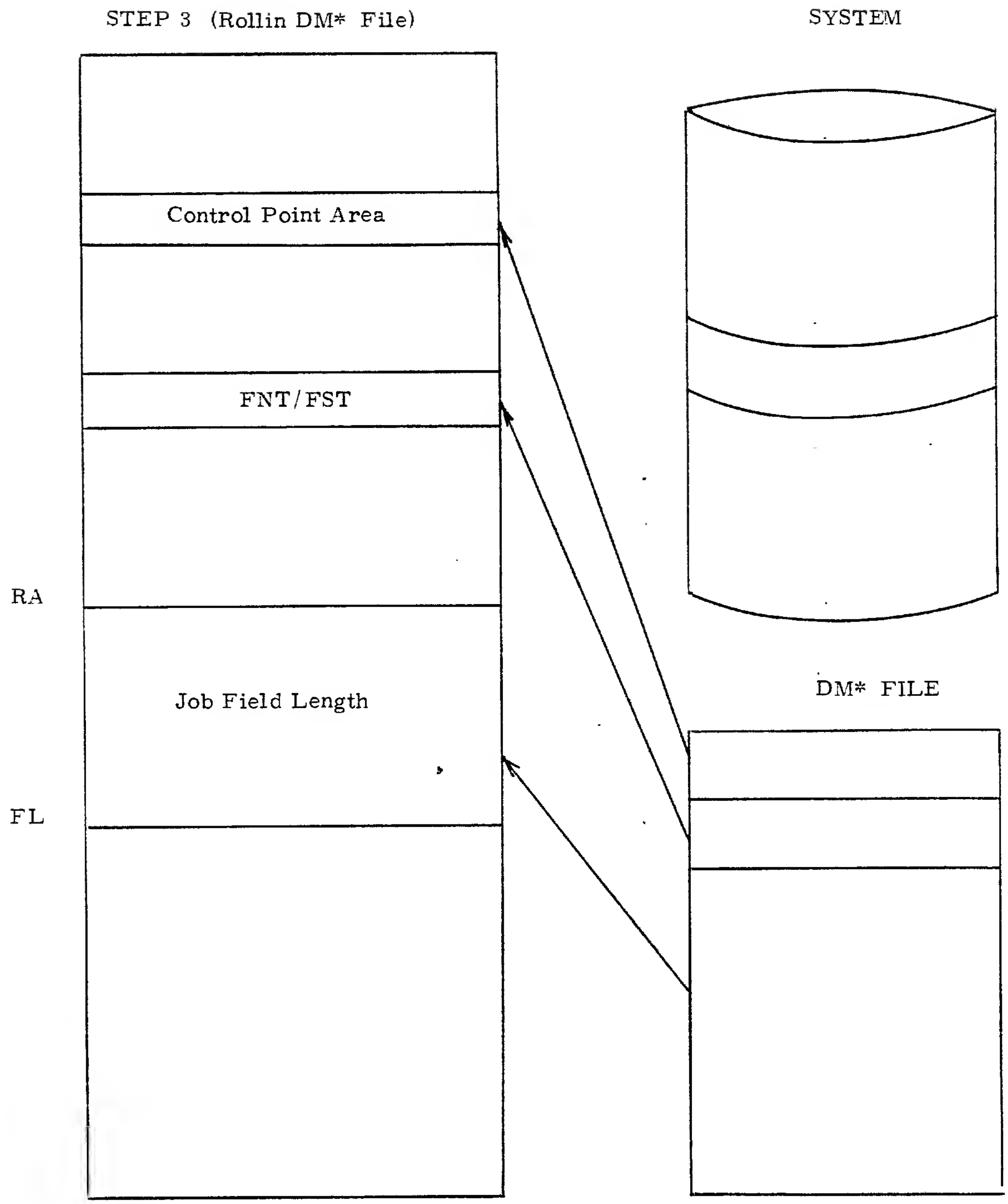


Figure 5-15. (1AJ Calls 1RI to Restore the Job)

DMP = JOB FLOW

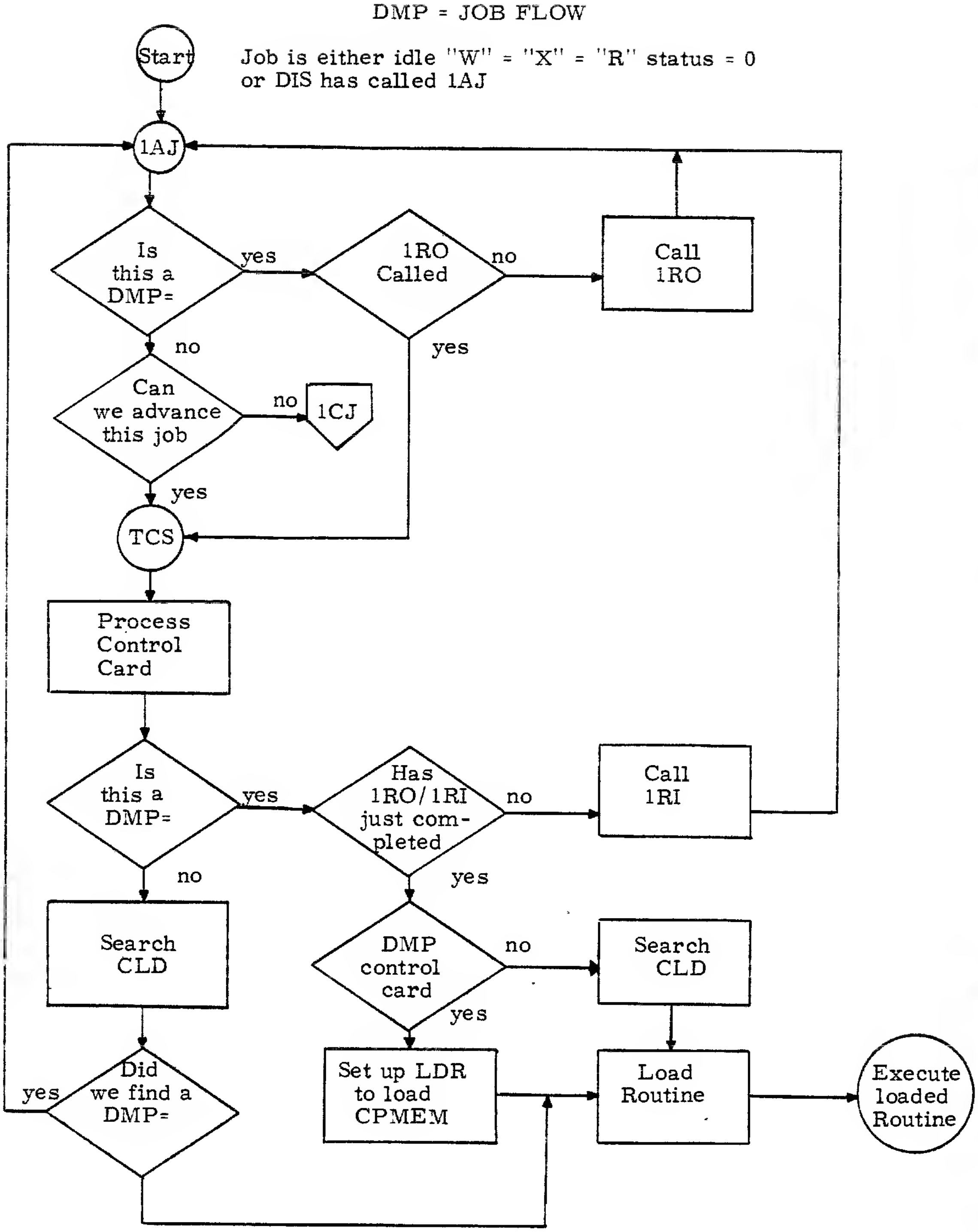


Figure 5-16. General Flow

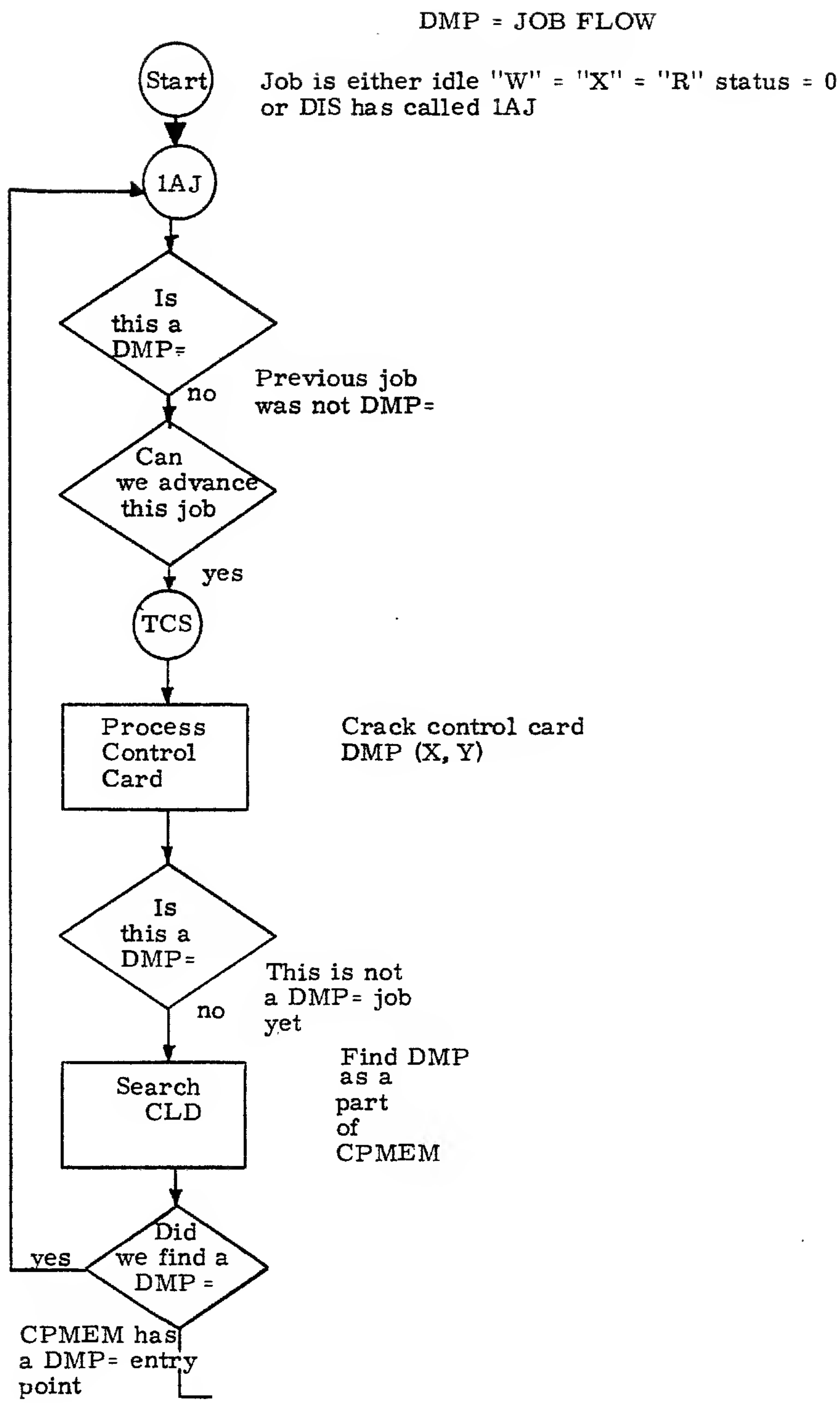


Figure 5-17. PASS 1 (Job Flow Has Come to a DMP Control Card)

DMP = JOB FLOW

Job is either idle "W" = "X" = "R" status = 0
or DIS has called 1AJ

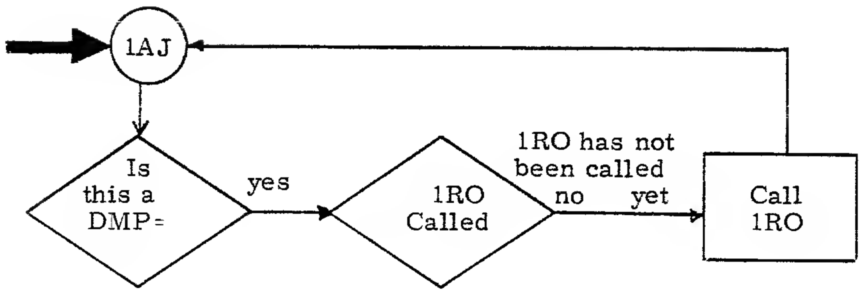


Figure 5-18. PASS 2

DMP = JOB FLOW

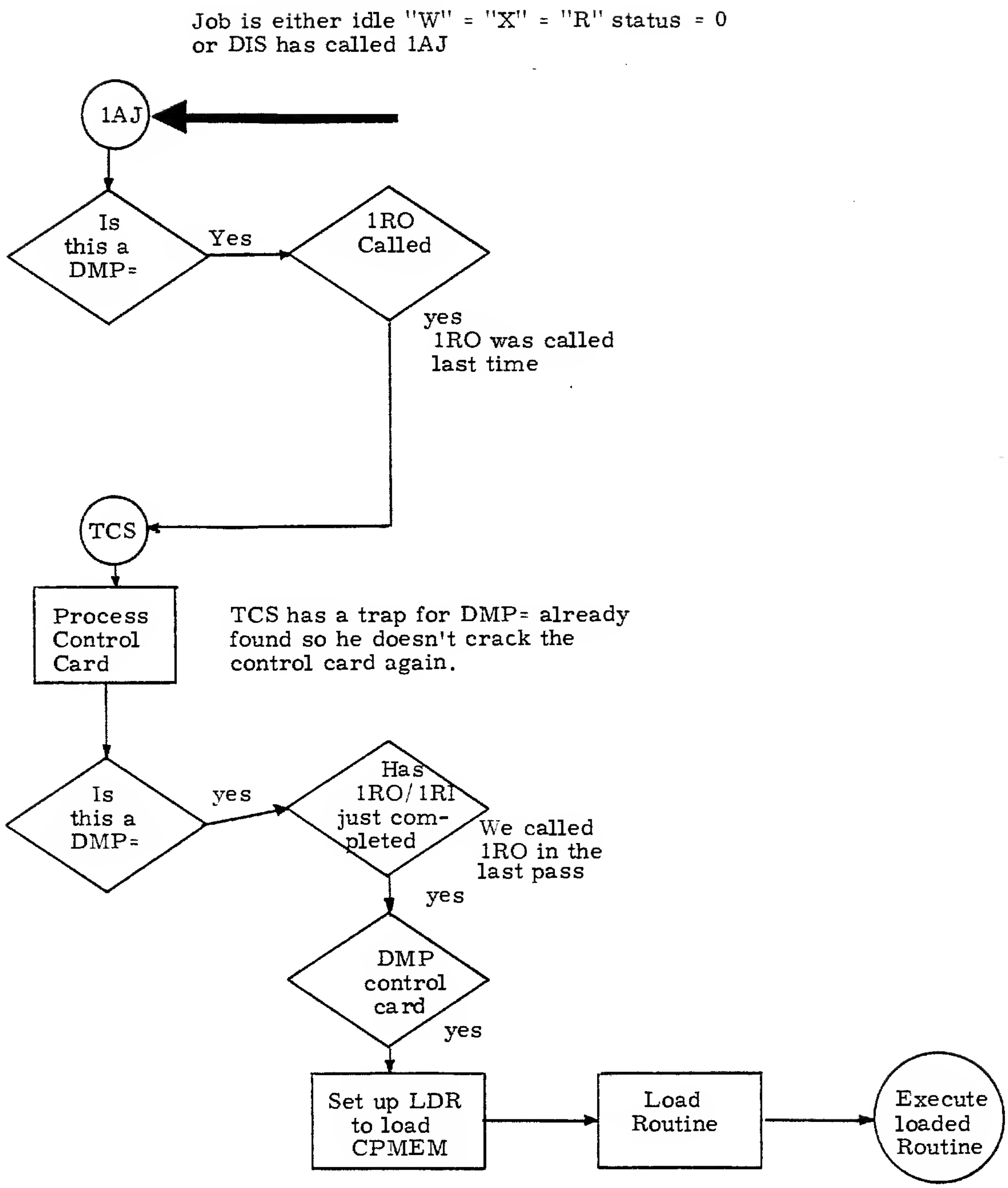


Figure 5-19. PASS 3

DMP = JOB FLOW

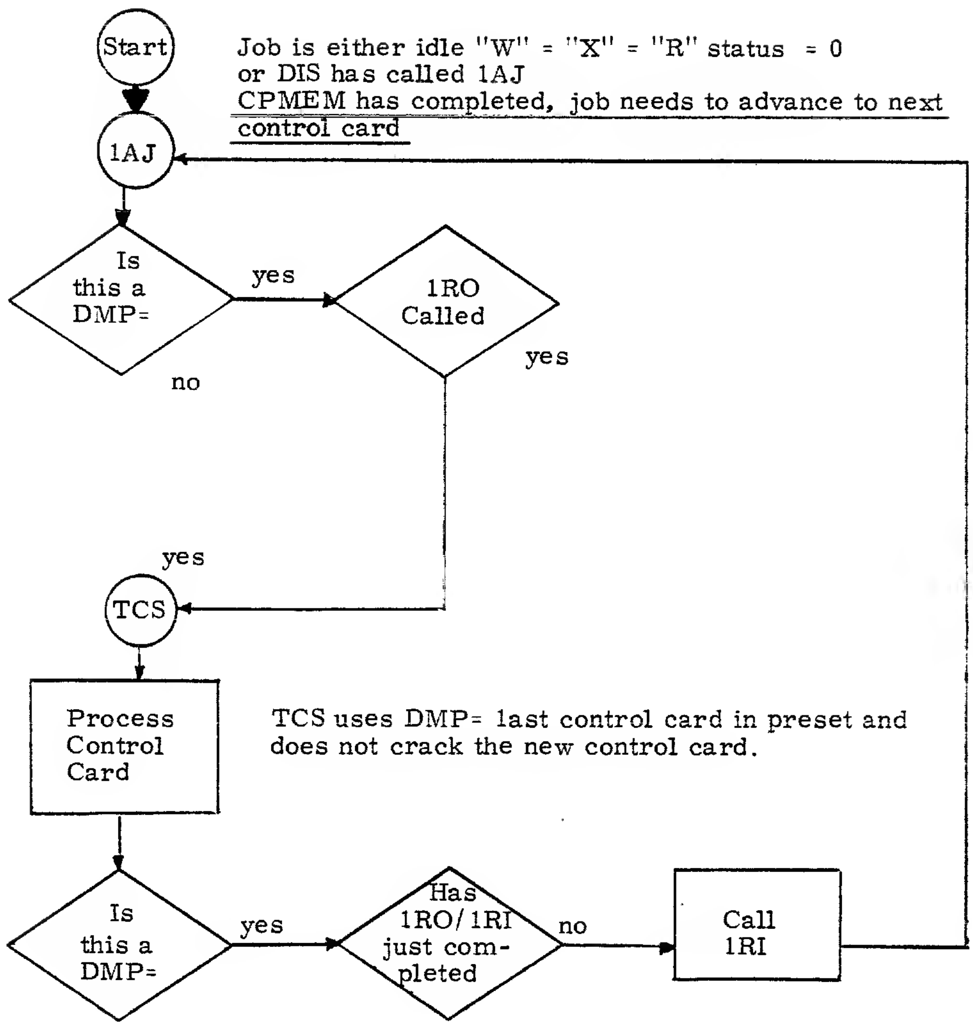
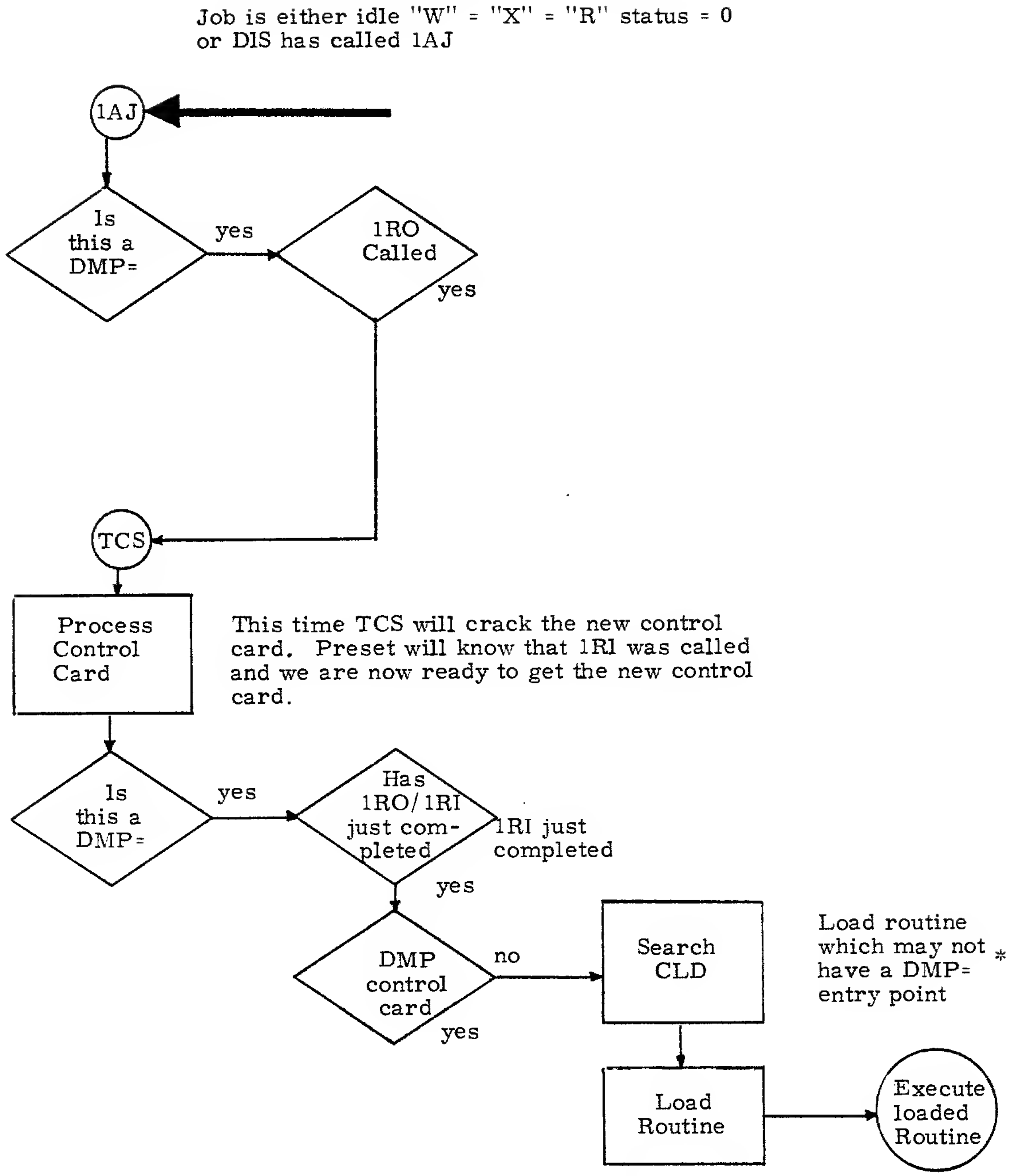


Figure 5-20. PASS 4

DMP = JOB FLOW



* An SEA job cannot initiate another SEA job.

Figure 5-21. PASS 5

6) SSJ=

Programs with SSJ= entry points are defined as special system jobs. The address specified by the SSJ= entry point, determines the start of a parameter area where the user accounting control words from the control point area are temporarily stored to allow the special system job access beyond the user's validation. When the special system job completes (or aborts) the user's validation parameters are retrieved from the parameter area within the special system job's field length and restored to the control point area. All local files created by the special system job (ID=SSID=74) are returned before normal control card processing is resumed. Whenever an SSJ= job creates a file, the FST ID field is set to SSID (EQU 74_8). In this way, 1AJ can ensure that any files attached to this control point during SSJ= processing will be released prior to returning control to the normal user.

a) A COMSSSJ common deck is provided to supply the calling program with special system job parameter equivalences.

b) An RFL= entry point must precede the SSJ= entry point to allow SYSEDIT to validate that the parameter area will fit within the special system jobs field length. If this condition is not satisfied, the SSJ= entry point will merely be added as another normal entry point for the program and no special processing will be done for it.

```
ENTRY RFL =  \  This is the only
ENTRY SSJ=   /  acceptable order.
```

c) The first word of the parameter area (SPPS) is used to set the CPA values. If it is zero, the current values are retained. Limits for these values are:

- $0 \leq$ CPU priority $\leq$ 70B
- $0 \leq$ queue priority $\leq$ MXPS+1
- $0 \leq$ time limit $\leq$ 77777B

Any other values are ignored. Thus, it can be ensured that a task does not get a time limit error, that a task has a higher CPU priority than a normal job, etc. Values are reset when the task terminates.

d) The SSJ= parameter block format is:

		24	12	12
SPPS		Time Limit	CPU Priority	Queue Priority

	42	18
UIDS	User Number	User Index

	12	12	12	12	12
APUS	Maximum Mag Tapes Allowed	Maximum Removable Packs Allowed	Maximum MS Tracks Allowed	Maximum Local Files Allowed	Maximum Deferred Batch Jobs Allowed

```
                    CTEXT  COMSSSJ  - SPECIAL SYSTEM JOB PARAMETERS.

                    COMMENT COPYRIGHT CONTROL DATA CORP. 1973.

          ***       COMSSSJ - SPECIAL SYSTEM JOB PARAMETERS.
          *         W.R. SACKETT. 73/01/27.

MM                  BASE M
          **        FILES CREATED BY SPECIAL SYSTEM JOBS HAVE AN *DIS* ID
          *         ASSIGNED TO THEM.  UPON SPECIAL SYSTEM JOB TERMINATION,
          *         ALL SUCH FILES ARE RETURNED.
          *         SPECIAL ID CODES USED BY THE SYSTEM.


74        SSID      EQU    74     SPECIAL SYSTEM JOB ID
75        CBID      EQU    75     CHECKPOINT FILE ID
76        CKID      EQU    76     CHECKPOINT FILE ID
77        SOID      EQU    77     SPECIAL OUTPUT ID
70        IDLM      EQU    70     USER LIMIT FOR ID CODES


          **        SPECIAL SYSTEM JOB PARAMETER AREA EQUIVALENCES.
          *         ACCESS PARAMETERS TRANSFERRED FROM USER,S CONTROL POINT
          *         ARE GIVEN THE FOLLOWING REFERENCES RELATIVE TO THE JOB,S RA.
          *
          *         THESE VALUES ARE STORED IN  CM FOR SYSTEM JOBS CONTAINING
          *         SSJ= ENTRY POINTS.  THEY ARE STORED AT THE ADDRESS SPECIFIED
          *         BY THE ENTRY POINT AND ARE RESTORED TO THE CONTROL POINT
          *         AREA WHEN THE SPECIAL SYSTEM JOB TERMINATES.
          *
          *         IF THE FIRST WORD OF THE PARAMETER AREA IS DEFINED TO BE
          *         NON-ZERO BY THE SPECIAL SYSTEM JOB, THESE VALUES WILL BE
          *         SET IN THE CONTROL POINT AREA -
          *         VFD    12/0, 24/TIME LIMIT, 12/CPU PRIORITY, 12/QUEUE PRIORITY


5         SSJL      EQU    5      PARAMETER AREA LENGTH

0         SPPS      EQU    0      SPECIAL SYSTEM PARAMETER VALUES
1         UIDS      EQU    1      USER IDENTIFICATION (UIDW)
2         APUS      EQU    2      ACCOUNT PERIPHERAL USAGE (APUW)
3         ACUS      EQU    3      ACCOUNT CENTRAL MEMORY USAGE (ACUW)
4         AACS      EQU    4      ACCOUNT ACCESS CONTROL (AAW)
          5-52
```

```
      **         TIMED/EVENT ROLLOUT EQUIVALENCES.                              COMSSSJ   46
      *          SYSTEM DEFAULT TIMES FOR EVENT ROLLOUTS.                       COMSSSJ   47
                                                                                COMSSSJ   48
                                                                                COMSSSJ   49
454  CRT         EQU   5*60D       *ROLLOUT*  MACRO DEFAULT TIME               COMSSSJ   50
740  MRT         EQU   10*60D      *REQUEST*  MACRO DEFAULT TIME               COMSSSJ   51
360  ART         EQU   4*60D       *ATTACH*  MACRO DEFAULT TIME                COMSSSJ   52
M M              BASE  *                                                        COMSSSJ   53
                 ENDX                                                           COMSSSJ   54
```

	24	12	12	12
ACUS		Maximum CPU Priority Allowed	Maximum Time Limit Allowed	Maximum FL Allowed in 100B Units

	60
AACS	Account Access Control Word. (Each bit has special meaning)

The entire SSJ= block will be swapped with the CPA values unless word 0 is zero, if word 0 is zero, then just store the users CPA in the 5 word block. In any case, when the SSJ= completes, the 5 word block will be restored into the users CPA, thus the SSJ= program can and does place any values he sets in this block into the CPA.

In fact, that is the way that ACCFAM sets up the users Verification area in the CPA, and is the way that CHARGE canset the VAL= flag bit 17 in UIDW to off. Also, the swap allows the SSJ= program to specify UI = 377777B for accessing VALIDUX,RESEXVF, etc. New for level 5: If the SSJ= user specifies SSJ= EQU 0, then the swap does not occur at all, and all files created by the SSJ= user will not get ID = 75B, so that the files remain for the caller, but the job will get SSJ= privileges, SIC, RSB, etc. The only use is for LINK, which can create binary files and other type files which the caller needs, such as LGO.

7) VAL =

When validation is enabled, the system will abort any job of non-system (SYOT) origin which attempts to load and execute as the first control card, any routine which does not have a VAL = SEP. This is the method employed to check Validation. The first two or three cards of a job stream must be JOB, ACCOUNT. and CHARGE (if needed). ACCOUNT causes the loading of ACCFAM, and CHARGE causes the loading of CHARGE, both of which contain VAL = SEPs. The system will allow these routines to run, and assuming that they don't abort the job, they will enter this job stream into the system. Once they are done, then the VAL= system checking is no longer done for this job. If a user did not have an ACCOUNT card as the second card (say COPY) it will force a load of a routine without VAL= SEP, and the job would be aborted by the system.

8) Following is a chart of which programs are currently making use of the special entry points.

	MFL=	ARG=	RFL=	DMP=	SSJ=	SDM=	VAL=
a. CHKPT			X	X	X		
b. RESTART			X	X			
c. RESEX		X	X	X	X		
d. CPMEM	X			X	X		
e. CHARGE		X	X		X	X	X
f. ACCFAM			X		X	X	X

		MFL=	ARG=	RFL=	DMP=	SSJ=	SDM=	VAL=
g.	PROFILE		X	X		X		
h.	MODVAL			X		X	X	
i.	PFILES			X			X	
j.	PF Utilities			X		X		
k.	BLANK		X	X		X		
l.	SYSEDIT			X		X		
m.	TRANSIM			X		X		

5.3.4 Special RA+1 Requests. (TLX, SIC, RSB)

The following three RA+1 requests can only be used by an SS or issued to an SS. Two of them (SIC - RSB) can also be used by SSJ= or QP>MXPS type jobs.

One "TLX" is used to call special PP routines, the other two, SIC and RSB, are used for inter-control point communication.

5.3.5 Special PP Calls

A normal CP routine may only request PP routines whose name begins with a letter. This is a protective feature to keep normal jobs from accessing certain system PP routines. By convention, any PP routine which should be available to a user, and is coded is such a way as to keep from destroying the system if called by an improper request, all have a letter as the first character of their name. Other dangerous PP's have a numeric as the first character of their names.

Sometimes it is desirable for a CP routine to call a special PP, such as TELEX calling 1TA. The TLX RA+1 request was designed for his purpose.

TLX process special PPU request is:

RA+1 =

TLX		addr

Where:

1) Addr = address of a cell containing the name of the PP routine desired and its arguments.

2) Auto recall is not honored

3) If the addr word is cleared, the request was honored and the PP routine was started.

4) If the addr word is unchanged when the CP regains control, the PP routine was not started (possible PP saturation, etc.).

5) The call is only honored for jobs whose QP is greater than MXPS. All other jobs will be aborted. Actually, the call is passed to a PP which will abort the CP unless a PP exists named TLX.

addr =

PP routine Desired	0	Arguments
18	6	36

5.3.6 Inter Control Point Communication For SS

The control point concept allows each control point to run independently of any other control points in the system. In addition each control point is protected from any other control point destroying any part of its CM FL. In some cases, however, it is necessary for one control point to communicate with another, as in TELEX, to TRANEX and RESEX to MAGNET.

An SS wishing to communicate with some other control point (maybe another SS) by sending information, can set up a communication block using ICAW in CPA and transfer it to a designated control point. Also, an SS may receive a block of data from some other control point (which may also be another SS).

The control of the transfer is based on the SS QP, and, therefore, must be unique. The buffers are defined in ICAW in CPA. The following two RA+1 requests are used for this communication.

- SIC – Send inter-control point data block from a control point program to the specified SS.

(RA+1) =

SIC	r		buff	St

SIC	-	Display code
r	-	1 if auto recall is desired in bit 40
buff	-	Address of the buffer to transfer to the subsystem
St	-	Address of status word for the transfer

(St) =

18	12	30
bn	sqp	reply

bn	-	Buffer number of subsystem to transfer to
sqp	-	Destination subsystem queue priority
reply	-	Not used

A block starting at buff will be moved to the indicated subsystem. The block length is specified in bits 0 - 17 of the first word of the block (buff), which includes this header. The block length must be less than 101B (to force CPUMTR in MTR mode, this operation must be very fast.).

NOTE

The request is honored only from jobs with: queue priority $\geq$ MXPS (i.e., sub-system status), or an SSJ= entry point defined (see 1AJ Section 6), or with access bit (CSTP (user may access special transaction functions) turned on.

The request from any other job will be treated as a PP call.

Response

	18	12	30
(ST) =	bn	sqp	reply

bn and sqp unchanged

reply - 1 If transfer completed successfully

3 If designation subsystem is not present in the system

5* If subsystem buffer is full, subsystem being moved, or subsystem job is advancing

7 If block length as specified in the first word is larger than that permitted by the subsystem.

11 If destination buffer is undefined by the subsystem

The format of the buffer block to transfer is:

buff + 0 =	0	block length = n - 1
+ 1	1st data word	
+ 2	2nd data word	
.		
.		
+n-1	n-1st data word	
+ n	nth data word	

NOTE: $n \leq 100B$ so entire block length is 101B.

- RSB – Send inter-control point block from SS to the calling control point, if no SS is specified, from absolute CM. The calling routine must have an SSJ= entry point defined (See special entry point, Section 5.2.2).

The format for this call is:

	18	6	6	12	18
(RA+1) =	RSB	r	0	sqp	St

Where:

RSB - Display code request

r - 1 if auto recall desired in bit 40

sqp - Subsystem queue priority (or control point to read). If it = 0, then block is read from core memory or relative to callers control point area. (see note on buffer below).

St - Address of status for the read.

* If auto recall is specified, the control point will remain in recall until condition 5 ends. The subsystem may indicate whether its buffer is full by setting the first word in the buffer non-zero, i.e., if the first word of the buffer in the SS is non-zero it can not receive info., if it is zero, it is ready to receive data.

	12	12	18	18
(St) =	0	WC	addr	buff

Where:

WC - Number of words to read

addr - Address to read from CM or buffer address relative to SS.

buff - Address of buffer to receive data in this CPs F1.

When sqp = 0, the contents of buff determines whether the read is from absolute CM or relative to the callers control pont area.

If (Buff)< 0, the read is from absolute CM and addr in the St word is the absolute address in CM to begin the read.

If (Buff) $\geq$0, the read is relative to the callers control point area, and Buff contains a list of addresses located within the CPA which are to be read. The list ends at WC or a zero list entry. The contents of the CPA address read is stored in the buff location which contains that address.

(Buff + 0) is mearly a flag denoting a read from absolute core or relative to CPA in the case whre sqp = 0. The calling program must have an SSJ= entry point.

Response

St =	Reply	WC	addr	buff

WC, addr, buff are unchanged

Reply = 4000_8 transfer completed successfully
2000_8 subsystem not present

If sqp $\neq$ 0, just fill buffer

If sqp = 0, and (BUFF + 0)<0, fill BUFF from absolute core as specified in addr field.

If sqp = 0, and (BUFF + 0) $\geq$ 0 (CPA read), then an example of this format follows:

Buff +0	+1
+1	STSW
+2	STSW-17B
+WC-2	MS1W
+WC-1	APJW

NOTE: Buff through Buff+WC-1 is WC words

then (job status word) from CPA will be stored in Buff+1

(2nd word of EPA) from CPA will be stored in Buff+2

(1st msg Buffer area) from CPA will be stored in Buff +WC-2

(Prog number area) from CPA will be stored in buff+WC-1

NOTE

Buff through Buff+WC-1 is WC words. It is not possible to get the first word of the EPA since the address would be 0 relative to CPA and any 0 word ends the list. It would be necessary to know the absolute address of the CPA to get the first word of the CPA.

The above is an example and is not intended to imply that only the CPA areas shown can be read.

JOB FLOW 6

6.0 INTRODUCTION

This section describes the detail job flow for 1SJ, 1AJ, 1CJ, 1RO and 1RI.

6.1 1SJ - JOB SCHEDULER

1SJ scans the FNT/FST looking for files of type input (INFT) or type Rollout (ROFT). 1SJ determines priorities for these entries via 1SP. 1SJ builds nine tables which it uses to determine which of the jobs in the input or rollout queue based on priority are to be reassigned to a control point and restarted. 1SJ rolls out any jobs which have a lower priority than this best job. It attempts to start the best job. If all fails, and 1SJ cannot find a best job to start, or cannot get enough resources for this best job, then 1SJ gives up.

The next time 1SJ is called, the best job may not be the same one picked the last time. A best job is only guaranteed a start up of the resources necessary are available at the time the job is being prepared.

MTR, 1AJ, 1RO, 1RI, 1TA or 1CJ may call 1SJ with the monitor function RSJM.

CPUMTR will check the Scheduler Active Flag (SAF) — JSCL+1 in CMR. If the SAF is clear, CPUMTR will call 1SJ to a PP. If the SAF is set, CPUMTR will also set the Scheduler Requested Flag (SRF) (JSCL in CMR).

When 1SJ completes a cycle, it will check the SRF. If it is clear, it will drop from the PP. If it is set, it will check the scheduler cycle (JSCL in CMR). If the cycle is less than some preset number, it will restart; otherwise, it will drop from the PP.

1SJ works with the current system status. Whenever many jobs make changes, these changes will affect 1SJ only while it is executing. The JSCL and JSCL+1 words ensure that only one 1SJ can run at any time in the system, and (with the cycle count) 1SJ will only run so many times before exiting. This ensures that the system is not constantly scheduling jobs in and out or assessing priorities too often and thereby wasting computer resources.

1SJ TABLES

1) TACP - Active control points

Entry = 1 word

Descending priority

Terminated by 0 entry

1	1	5	5
P	R	0	CP

P – Rollout in process

R – Rollout requested (used only in CFL) (Commit FL)

CP – Control point number

2) TRST - Table of rollout status

Entry = 1 word

Indexed by control point number

1	1	10
P	R	0

P – Rollout in process

R – Rollout requested (used only in CFL)

3) TJFL - Job field length

Entry = 1 word

Indexed by control point number

12
FL

FL – Field length assigned at control point

4) TAFL - Available field length by control point

Entry = 1 word

Indexed by control point number

1	11
A	FL

A – Control point assigned to job

FL – Field length available at control point

5) TJPR - Job priority

Entry = 1 word

Indexed by control point number

12
PR

PR – Priority of job

6) TJOT - Job origin type

Entry = 1 word

Indexed by control point number

Set only if job active

12
OT

OT – Origin type of job

7) TMFO - Table of total available field length for all jobs of an origin type

Entry = 1 word

Indexed by origin type

12
FL

FL – Field length available

8) TAFO - Table of assigned field length by origin type

Entry = 1 word

Indexed by origin type

12
FL

FL – Field length assigned

9) TMJO - Table of maximum field length per job by origin type

Entry = 1 word

Indexed by origin type

12
FL

FL Maximum FL allowable for a job

The call:

18	6	36
1SJ	CP	0

programs called:
- 1AJ - Advance Job Status
- 1RI - ROLLIN File
- 1SP - EVALUATE Priorities

A diagram of the 1SJ interaction follows:

RSJM MTR FUNCTION

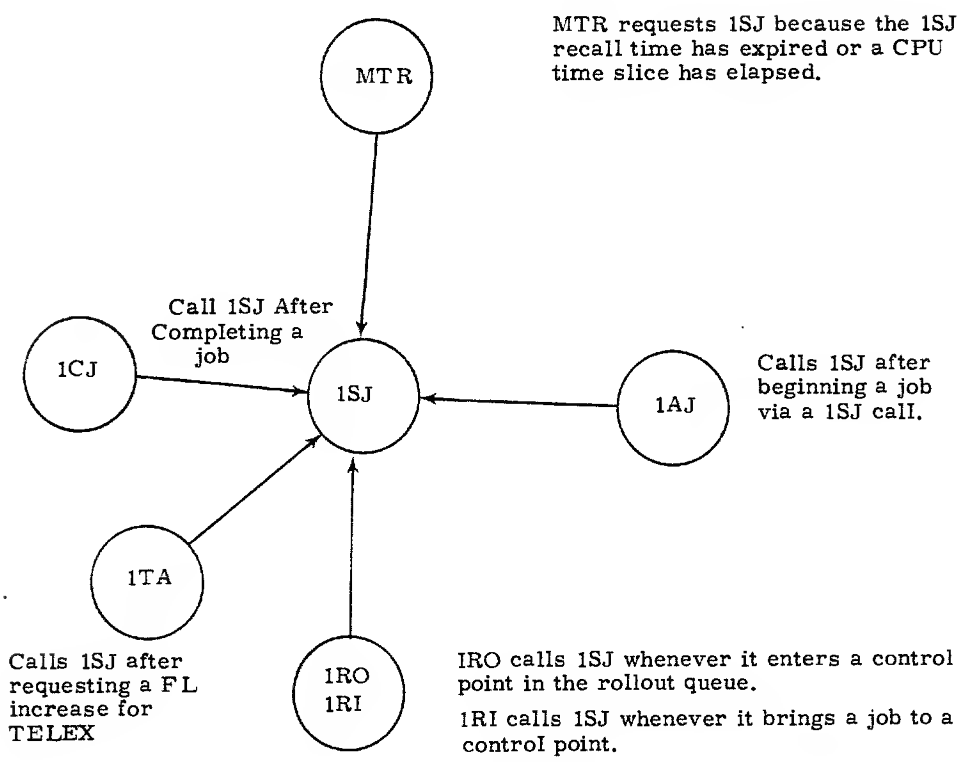
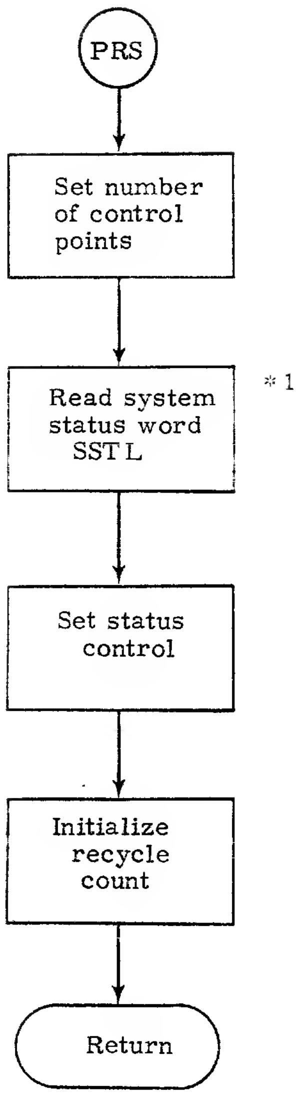


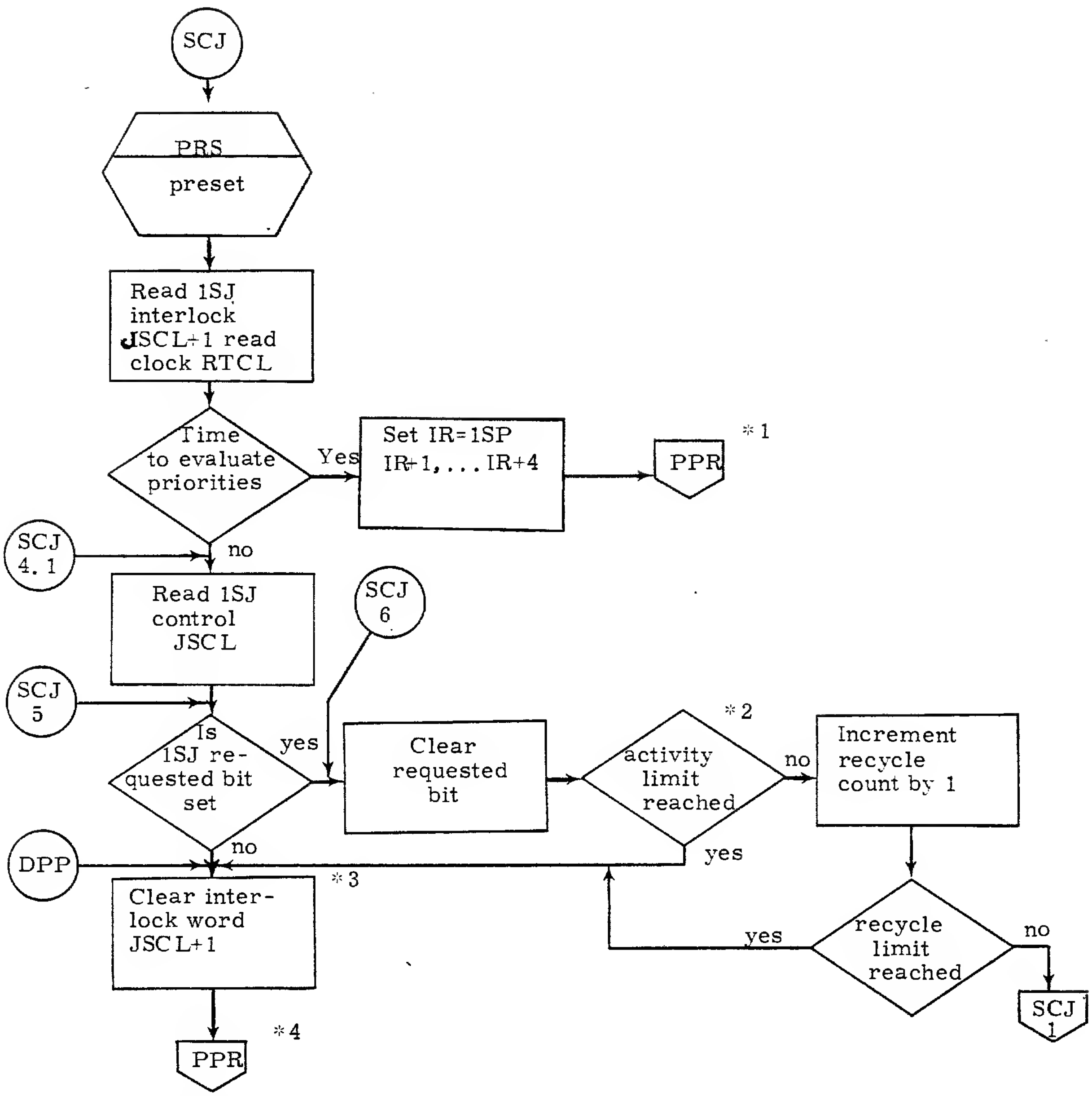
Figure 6-1. 1SJ Interaction

A flowchart of the 1SJ main loop and 1SP main loop are diagrammed in Figure 6-2 through 6-4.



*1 disable/enable control

Figure 6-2. 1SJ PRS – Normal Preset (Non-Subsystem Job)



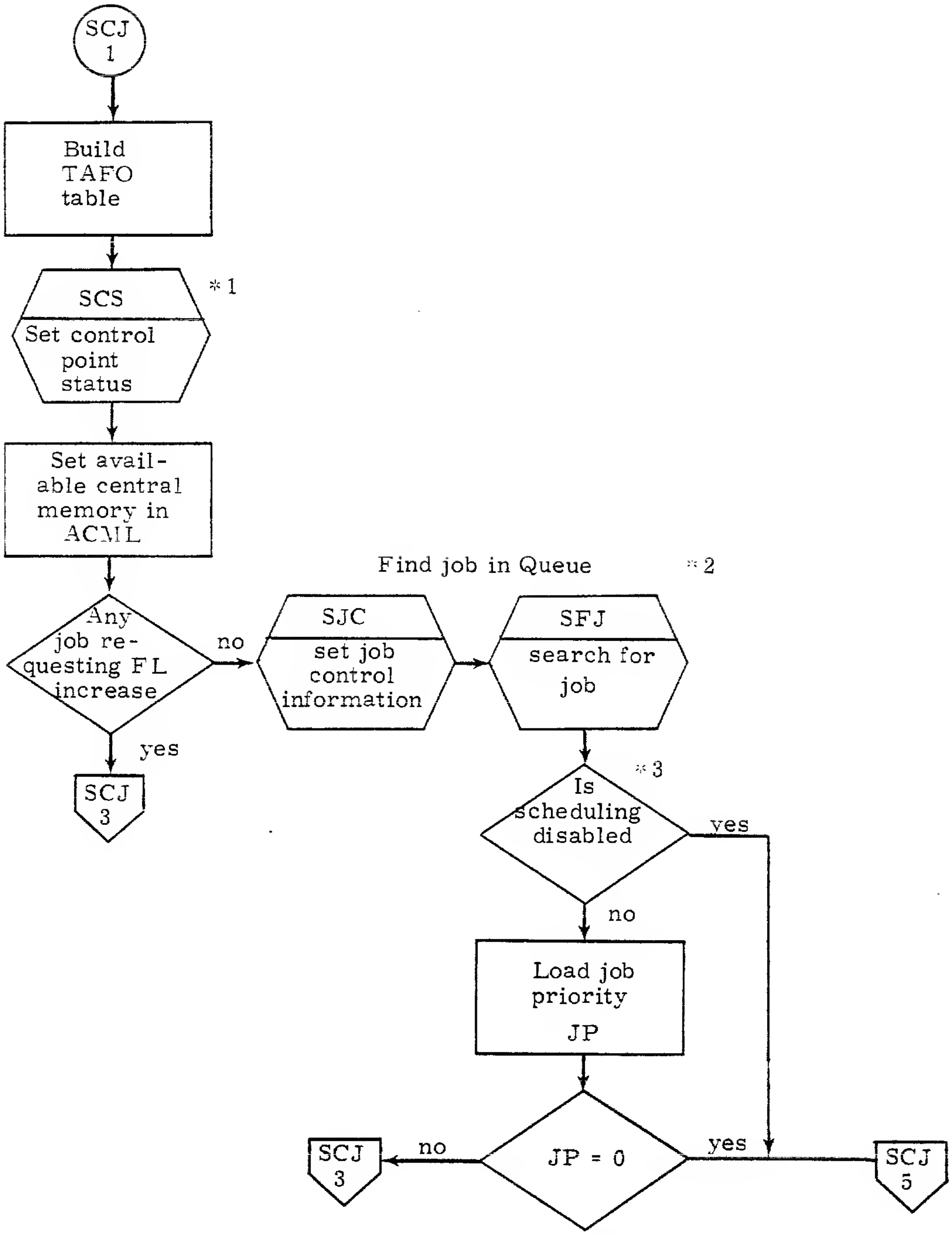
* 1 1SJ call 1SP on top of itself, when 1SP is done he will call 1SJ back to this PP. (This interchange is done without informing monitor.

* 2 If maximum number if rollouts reached don't issue anymore.

* 3 Whenever a routine calls 1SJ it uses the RSJM function CPUMTR will set the interlock bit SAF and call 1SJ, but if its already set CPUMTR will ignore the request telling the PP that 1SJ was called and CPUMTR sets the SRF.

* 4 Drop the PP.

Figure 6-3. 1SJ Main Program



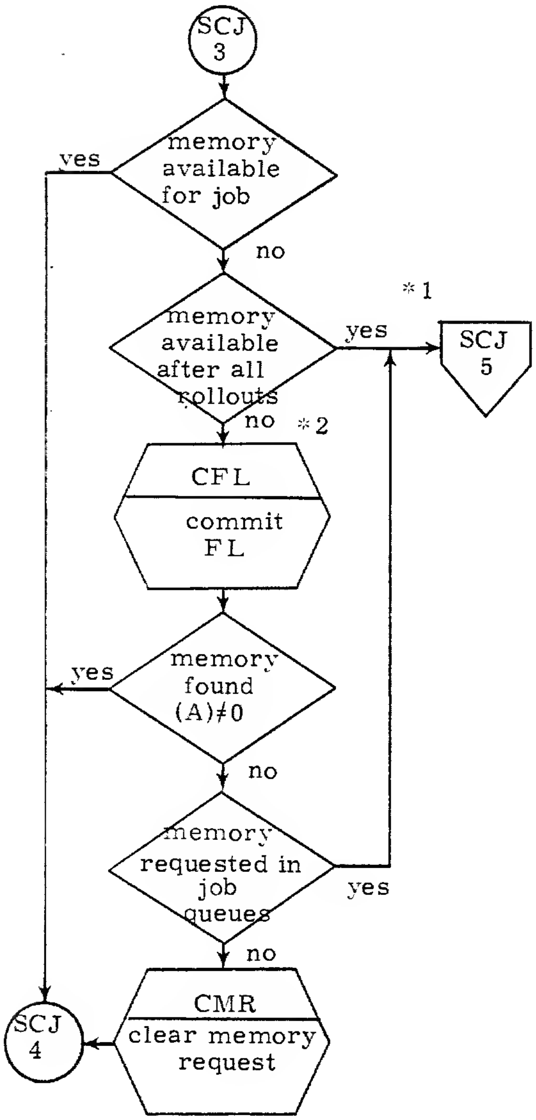
*1 Take snap shot of control points and build RACP, TRST, TJFL, TAFL, TJPR and TJOT tables. These tables are used by 1SJ to make all scheduling decisions.

*2 SFJ finds a job to schedule and determines if it is a subsystem job or regular job.

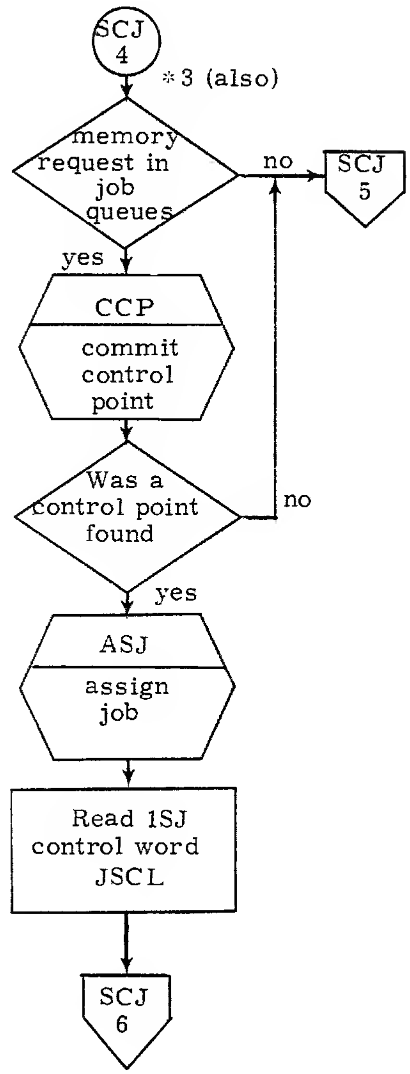
*3 Since requested flag cleared during last jump to SCJ this will end 1SJ.

Figure 6-3. 1SJ Main Program (Continued)

Check memory requirements
if central memory not available
rollout jobs to reclaim memory



Check control point assignments.
If control point not available,
rollout some other job.

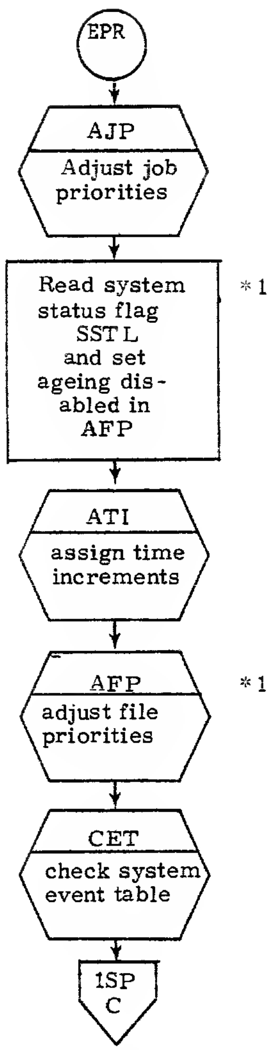


*1 If scheduled rollouts will clear up enough FL for job then 1SJ can go away and come back later when the rollouts are complete.

*2 Schedule rollouts until enough FL will become available. Use ROCM function for all control points whose priority is lower than the one we are trying to schedule.

*3 This is a switch telling 1SJ whether he is trying to schedule a job to a control point or attempting to increase a running job FL.

Figure 6-3. 1SJ Main Program (Continued)



1SP is called periodically by 1SJ to evaluate priorities of files in the INPUT and OUTPUT queues. The CM time slice parameters are checked for jobs at control points.

If CM time slice has elapsed and this job is within the queue ageing range, set this job priority = lower bound priority.

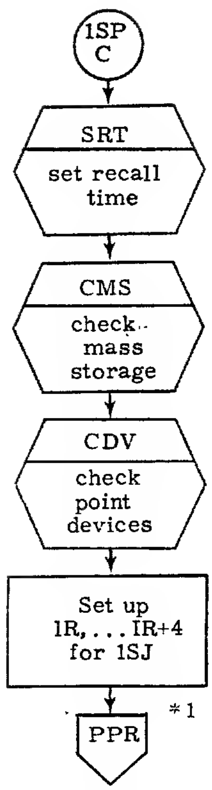
Add 1 to current interval count in JBC Job Control area Priorities

Adjust queue priorities of all files in INPUT or ROLLOUT queues whose priorities fall within the queue ageing priority range.

System event table is checked for resources described in (TEFT) time/event table FST entries encountered in AFPs search. He updates the time a job has sat in this table.

*1 If ageing disabled, AFP will not advance priorities in the FST of the INPUT and ROLLOUT QUEUES.

Figure 6-4. 1SP Evaluate Priorities



Set up the 1SJ control information JSCL in CMR. (Scheduler Cycle)

See if time to activate Label Check routine. See if any RMS initialize requests pending.

See if time to check point devices; if so, then copy MST information and TRT onto device (in case of system failure, devices can be recovered). CDV sets up a call to 1CK if devices are to be check pointed.

*1 Call 1SJ into this PP to continue. If 1SP found anything for 1SJ to do he will set the scheduler requested flag JSCL in CMR.

Figure 6-4. 1SP Evaluate Priorities (Continued)

6.2 1AJ — ADVANCE JOB STATUS

1AJ is called to advance the status of an active job. This action may be caused by:

1) The Job Scheduler (1SJ) wants to start a new job just scheduled to a control point.

2) Either monitor has sensed no activity at a control point ("W" and "X" bits clear).

3) "DIS" or other similar programs wish to process an error flag or a control statement.

The general 1AJ call is as follows:

- PPU Direct Cells

	IR	IR+1			IR+2	IR+3	IR+4
	18		1	5	12		
Input Register (IR) =	1AJ or	TCS	0	CP	FN	Parameter	

CP = Control point number

FN = Function number

P = Parameter

The (TCS) Translate Control Statement can be called directly.

- FN=5 (For Control Card Read and Execute)

18	1	5			
TCS	0	CP	5	0	ADDR

ADDR = address from which to read control statement

All other 1AJ calls are defined below:

- FN = 0 (From MONITOR, 1RO, and DIS for SSJ = and DMP=

	IR		IR+1		IR+2	IR+3	IR+4
	18		1	5	12	12	12
IR =	1AJ		0	CP	0	0	N

N = 1, set by 1RO upon completion of DMP= processing

N = 2, set by DIS for SSJ= and DMP= processing

- FN=1 (From 1SJ)

	18		1	5	12	12	12
IR=	1AJ		0	CP	1	0	FA

FA = address of input file

- FN=2 (From DIS)

IR =

	1	5	12	12	12
1AJ	0	CP	2	0	DC

DC = Bit 2 set, indicates take control statement from MS1W

Bit 1 set, indicates return error message to MS2W with no error flag on invalid control card.

Bit 0 set, indicates read statement and stop prior to execute (RSS indicator)

- FN=3 (From other PP programs)

IR =

18	1	5	12	12	12
1AJ	0	CP	3	0	0

- FN=4 (For control card fetch)

IR=

18	1	5	12	6	18
1AJ	0	CP	4	SF	ADDR

SF = Subfunction number for reading control statement
= 0 Advance pointers
= 1 Read only if not a local file load
Do not advance pointers
= 2 Set bit 17 in argument count if local file load
= 4X If parameters to be cracked in SCOPE format

ADDR = Address to READ/WRITE control statement FROM/TO

The dayfile message is "SPCW CALL ERROR". This signifies that a DMP= type call was made, and the program called is either not in the CLD or does not have a DMP= entry point defined.

The programs called by 1AJ are:

1CJ — Complete job
1RI — DMP= rollin
1RO — Rollout job, normal rollout and DMP= rollout
CIO — Complete special files on errors
DMP — Exchange package dump (for certain error flags)
RPV — Process reprieve errors (SCOPE function)

The common direct location assignments are:

Name	Value	Description
AB	20-24	Assembly buffer
CN	25-31	CM word buffer
FS	32-36	FST entry
EP	37	Entry point pointer
SP	40-44	Statement pointer
OT	45	Job origin type
EF	46	Error flag
RO	47	Rollout flag
FA	57	Address of FST entry
CW	60-64	Library control word
RF	65	Reprieve error flag

In general, 1AJ is called by MTR, 1SJ, or DIS. However, in the case of Special Entry Points (SEP) 1RO will call 1AJ back after rolling a job out to DM* and setting up a control point for the Special Entry routine. An SEP can be rolled out, and when it is rolled back in, 1RI will call 1AJ to advance it.

1AJ, 1SJ, MTR, 1RI and 1RO interaction are illustrated in Figure 6-5.

1AJ uses the following overlays:

1) 3AA — Begin job

2) 3AB — Process error

3) TCS — Translate control statement

4) LDR — Absolute CPU overlay loader

5) 3AC — Search peripheral library

6) 3AD — Search for overlay

7) 3AE — Load copy routines

8) 3AF — Special entry point processor

A description of each overlay and their flowcharts follow in Figures 6-6 and 6-7.

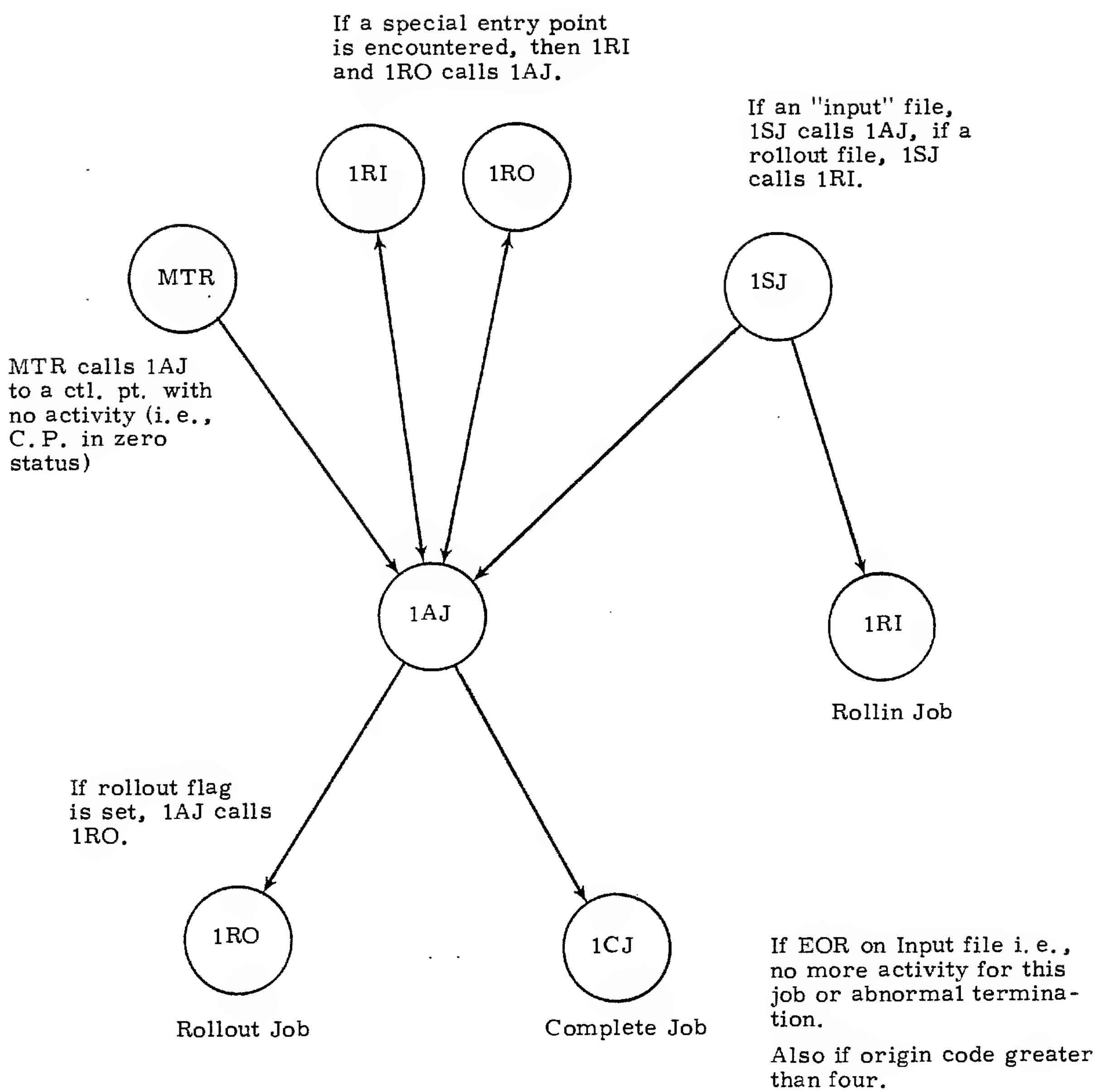


Figure 6-5. 1AJ Interaction

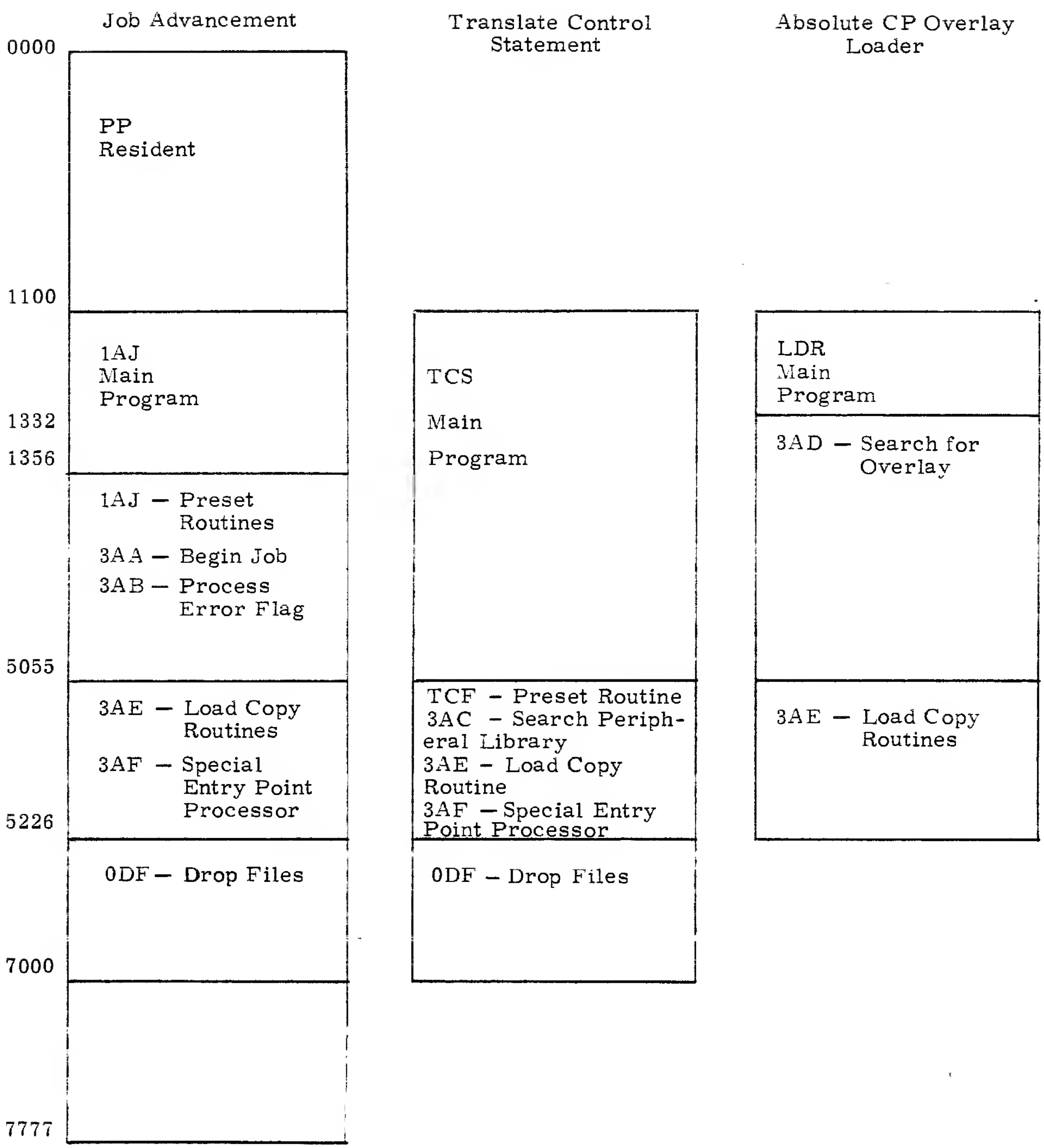
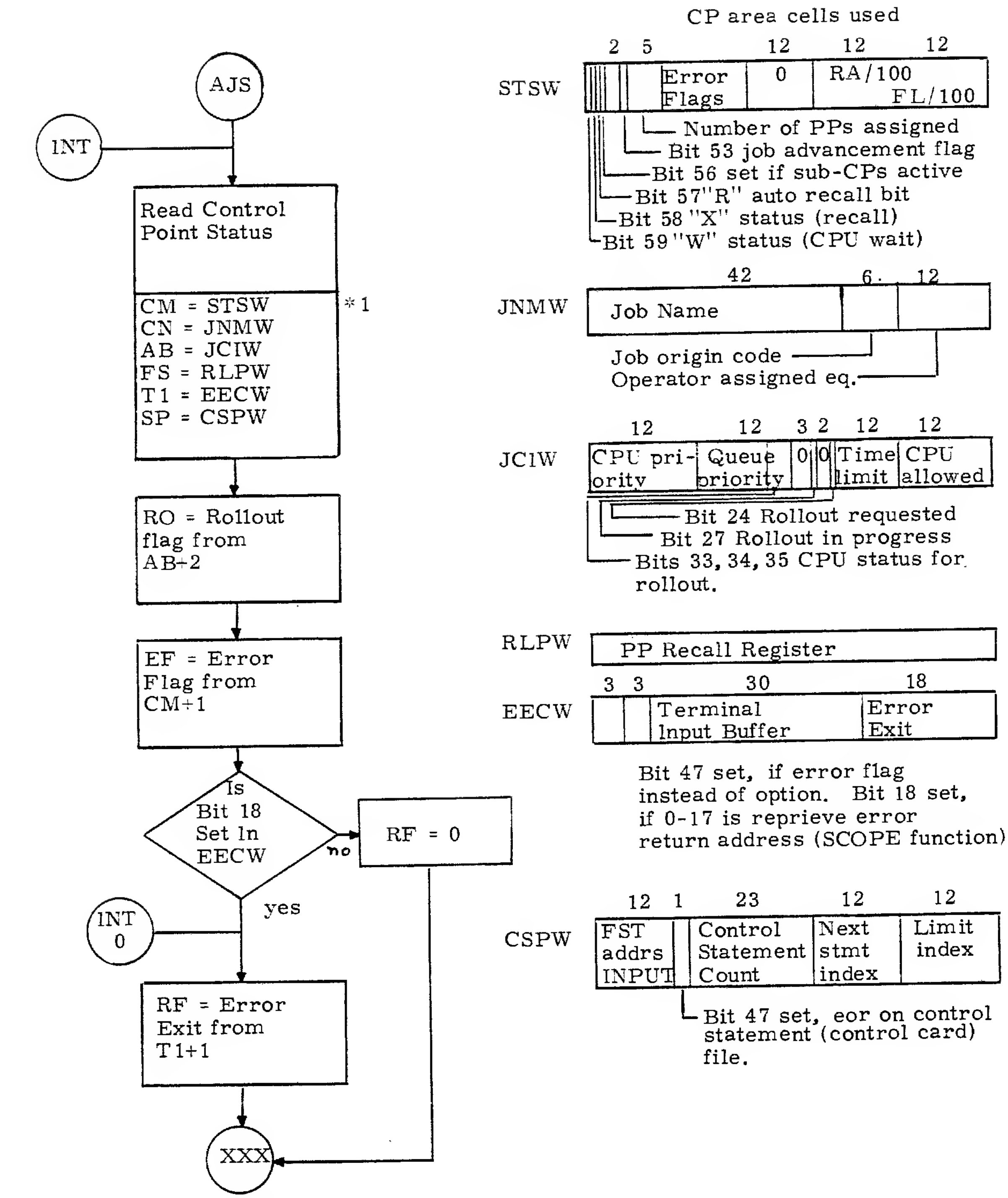


Figure 6-6. 1AJ Major Overlay Core Layout



*1 Read 1 CM word into 5 PP words

Figure 6-7. 1AJ — Advance Job

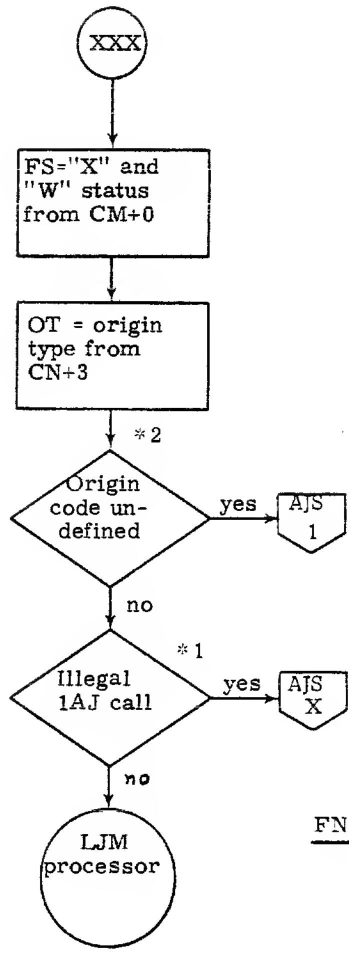


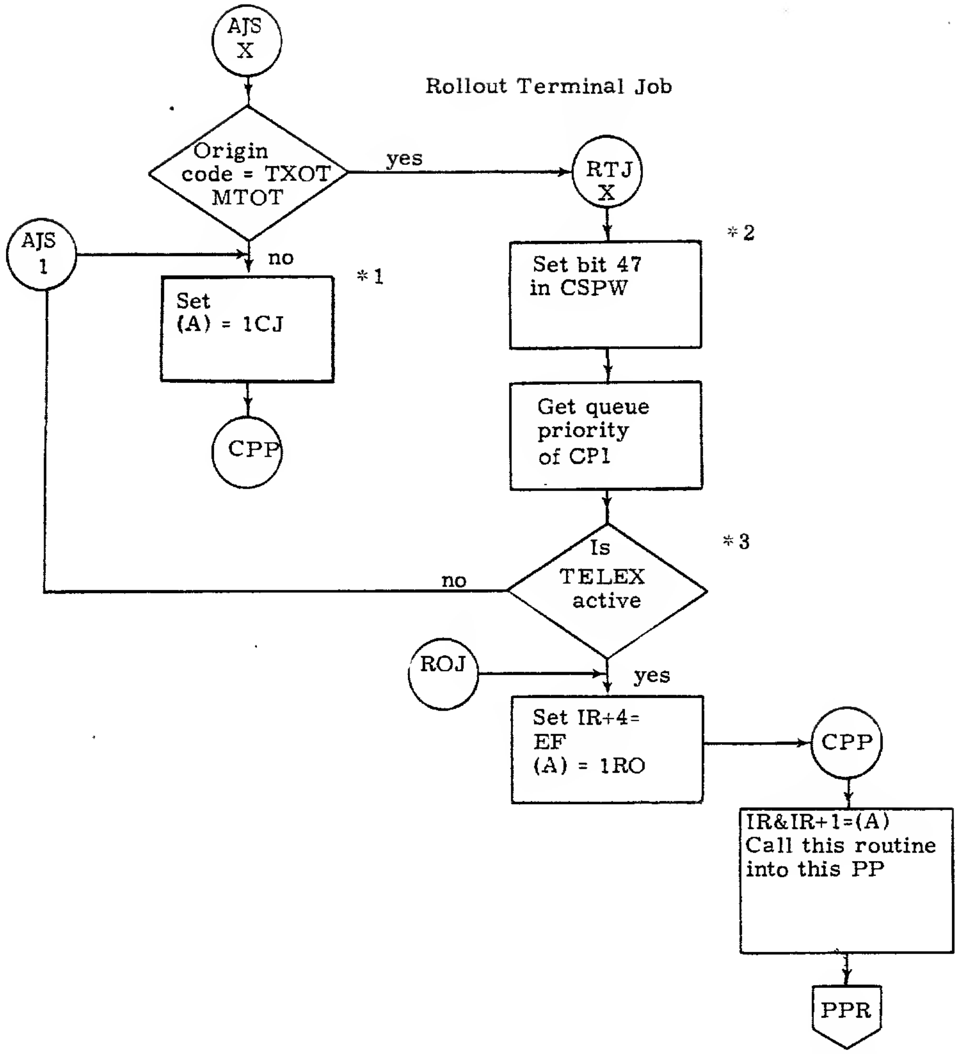
Table of Processors

FN Function #	Functions	Name
0	Monitor call	CSR
1	1SJ call	SCH
2	DIS call	CSR
3	other PP call	OPP
4	control card fetch	

*1 is FN function from IR+2 $\geq$ 5

*2 protective code. If an origin code > 4 is not trapped the processors will malfunction and the system could crash.

Figure 6-7. 1AJ – Advance Job (Continued)



*1 Since it is not TELEX origin and no advancement is possible terminate the job.

*2 Ensure empty control card buffer by indicating eor.

*3 Is queue priority of job at CP1 = 7775B

Figure 6-7. 1AJ – Advance Job (Continued)

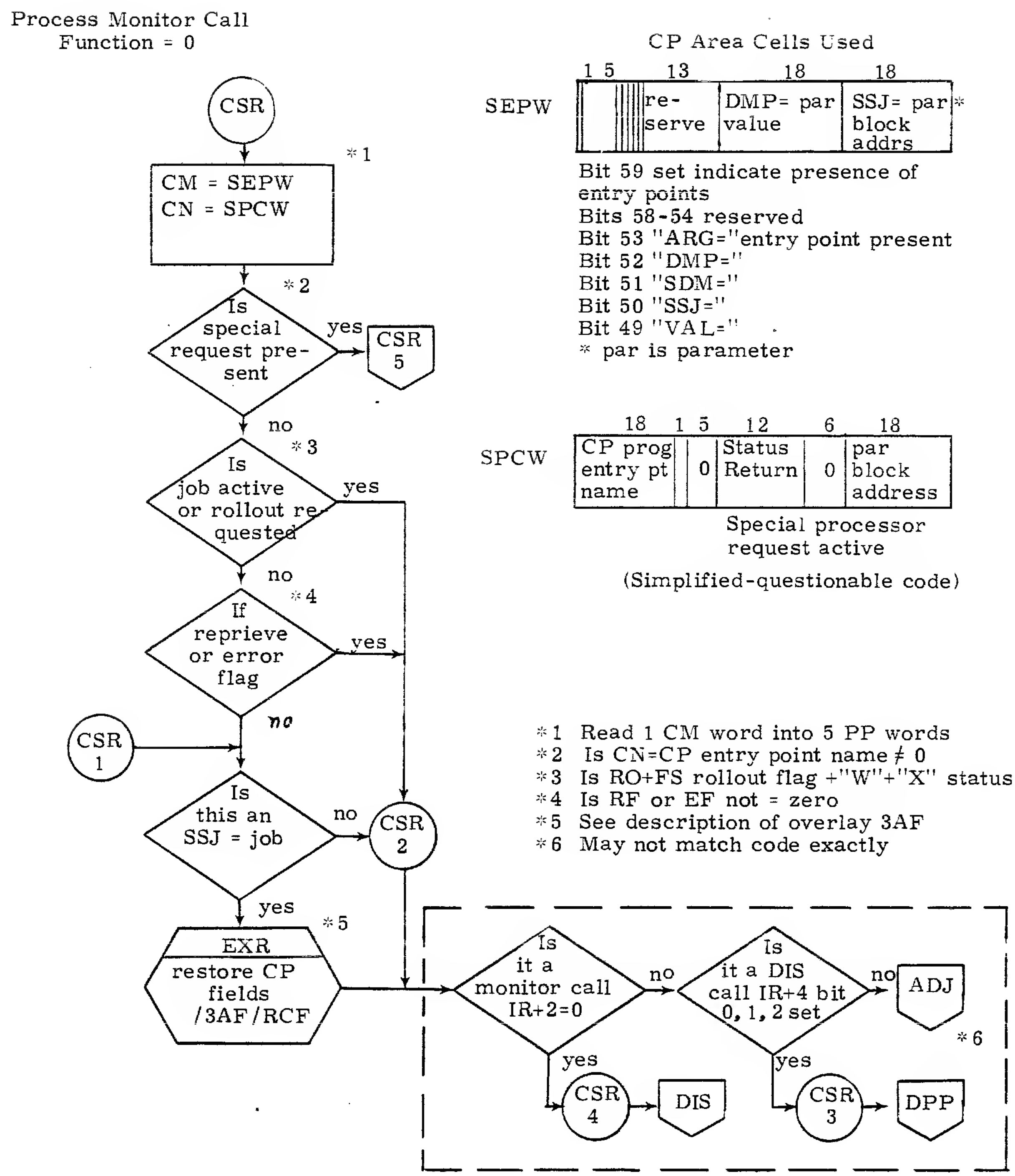
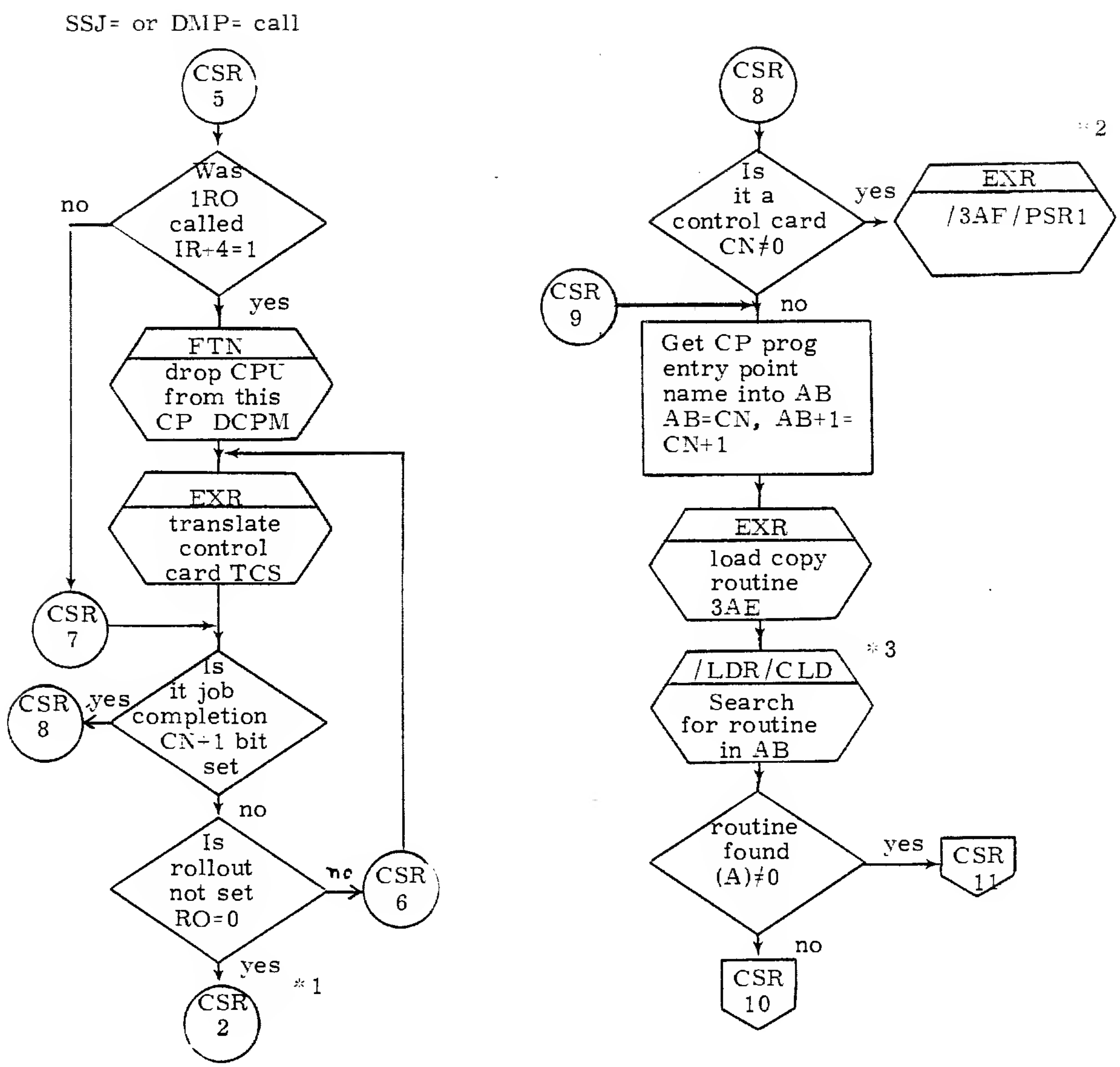


Figure 6-7. 1AJ – Advance Job (Continued)



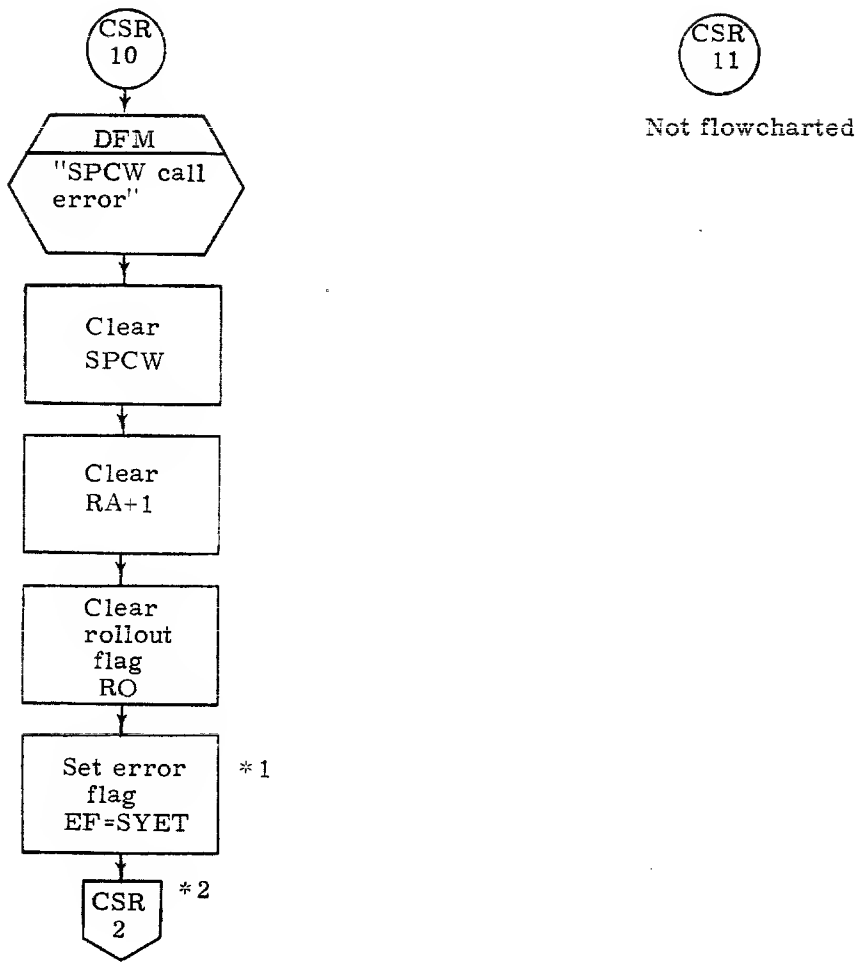
*1 This path will force job to be rolled out and 1AJ will drop.

*2 3AF exits via a call to 1RO and drops from PP.

*3 CLD is a routine which is loaded wherever *CALL COMPCLD is.
CLD searches the Central Library directory for the entry in AB. On exit:
(A)= address of Library Control word or =0 if not found.

Figure 6-7. 1AJ — Advance Job (Continued)

SSJ= or DMP= call (cont'd)

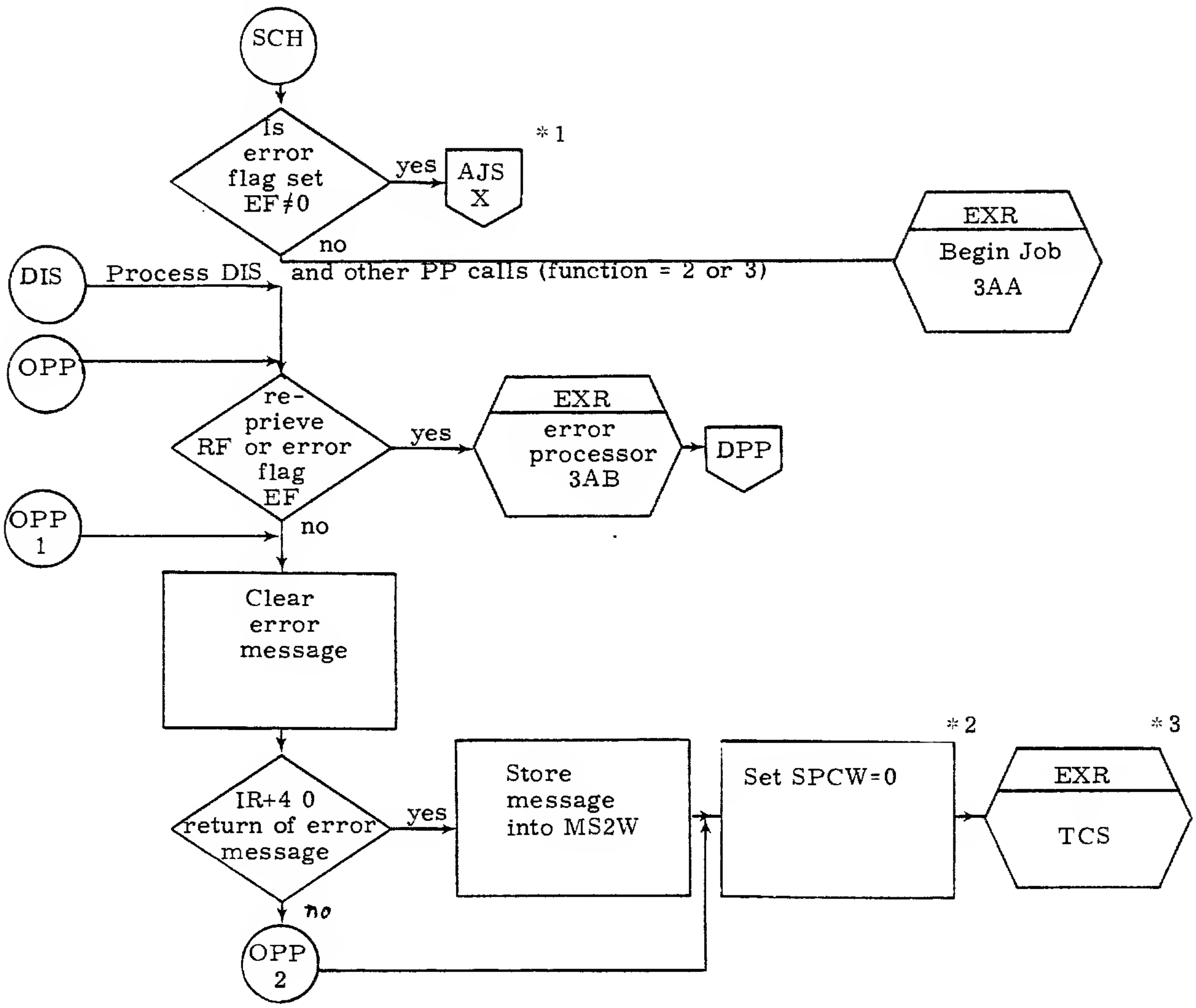


*1 System abort error

*2 1AJ will drop and this CP will be aborted.

Figure 6-7. 1AJ — Advance Job (Continued)

Process scheduler call (function = 1)



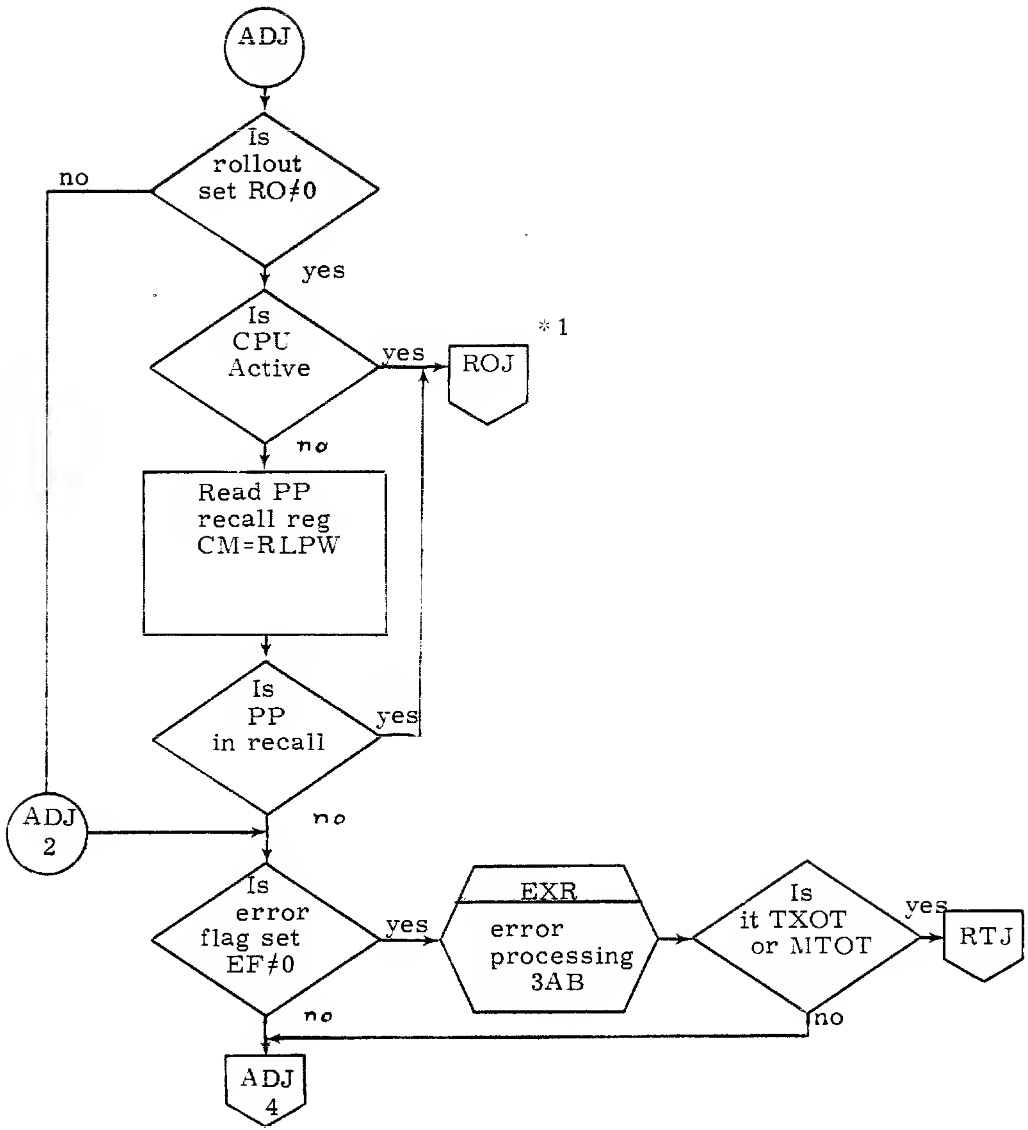
*1 Exit to 1CJ if error flag set

*2 Turn off any special processor commands

*3 Read next control card and advance the job. If illegal control card then abort.

Figure 6-7. 1AJ – Advance Job (Continued)

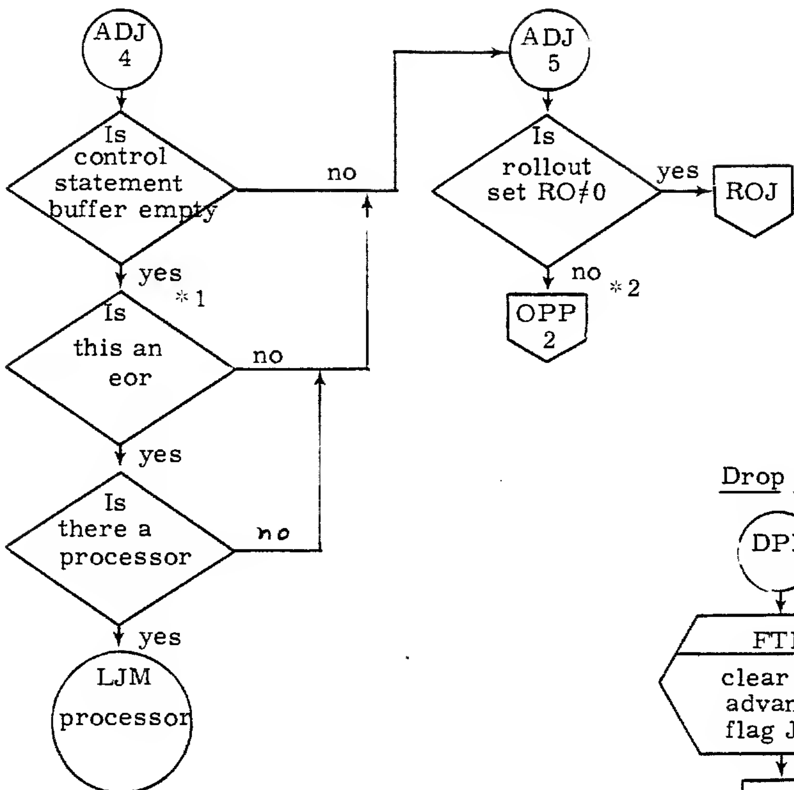
Advance job



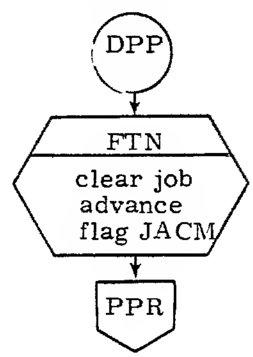
*1 Rollout file

Figure 6-7. 1AJ – Advance Job (Continued)

Advance job (cont'd)



Drop PP



Processors Type	Processor
SYOT	AJSX
BCOT	AJSX
EIOT	AJSX
TXOT	RTJ
MTOT	RTJ

*1 Is this the end of the control cards, then terminate

*2 Call TCS to process this card

Figure 6-7. 1AJ - Advance Job (Continued)

6.2.1 3AA — Begin Job

3AA initiates job processing at the control point.

The only dayfile message is:

JOB CARD ERROR

The direct location assignments are:

Name	Value	Description
PP	60	Pot pointer
TN	61	Terminal number
PA	62	Pot address (2 words)
TT	64	Terminal Table address (2 words)
TA	66	TELEX reference address

6.2.2 3AB — Process Error Flag

3AB processes error flags by sending an error message to the dayfile. In the case of an arithmetic error, a call is made to DMP to dump the exchange package area.

When these operations are complete, the control statement buffer is searched for the control statement EXIT. If this statement is found, 3AB returns to 1AJ to continue statement processing. If an EXIT is not found, control returns to 1AJ to complete the job processing.

The dayfile messages are:

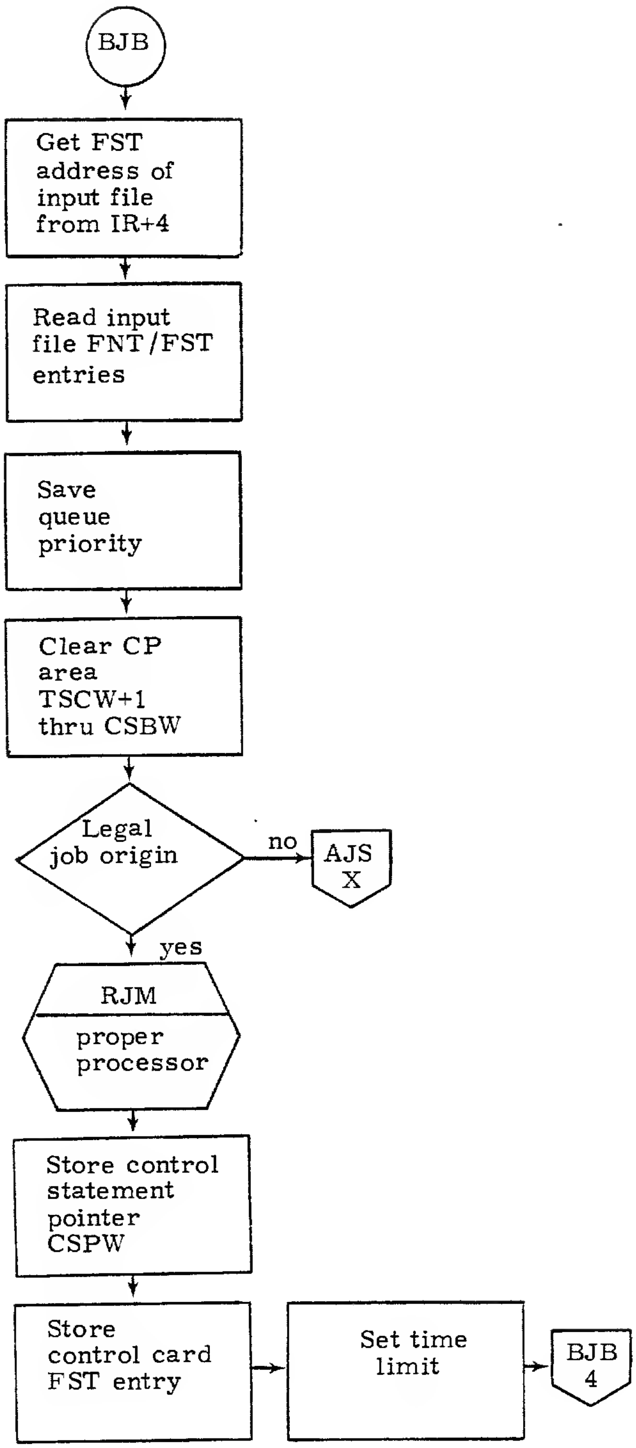
1) "TIME LIMIT." = The monitor has detected that the time limit for the job has expired.

"ARITH, ERROR x AT yyyyyy." = The monitor has detected an arithmetic error condition x at address yyyyyy.

"PP CALL ERROR." = The monitor has detected an error in a CPU request for PP action.

"OPERATOR DROP." = The operator has dropped the job.

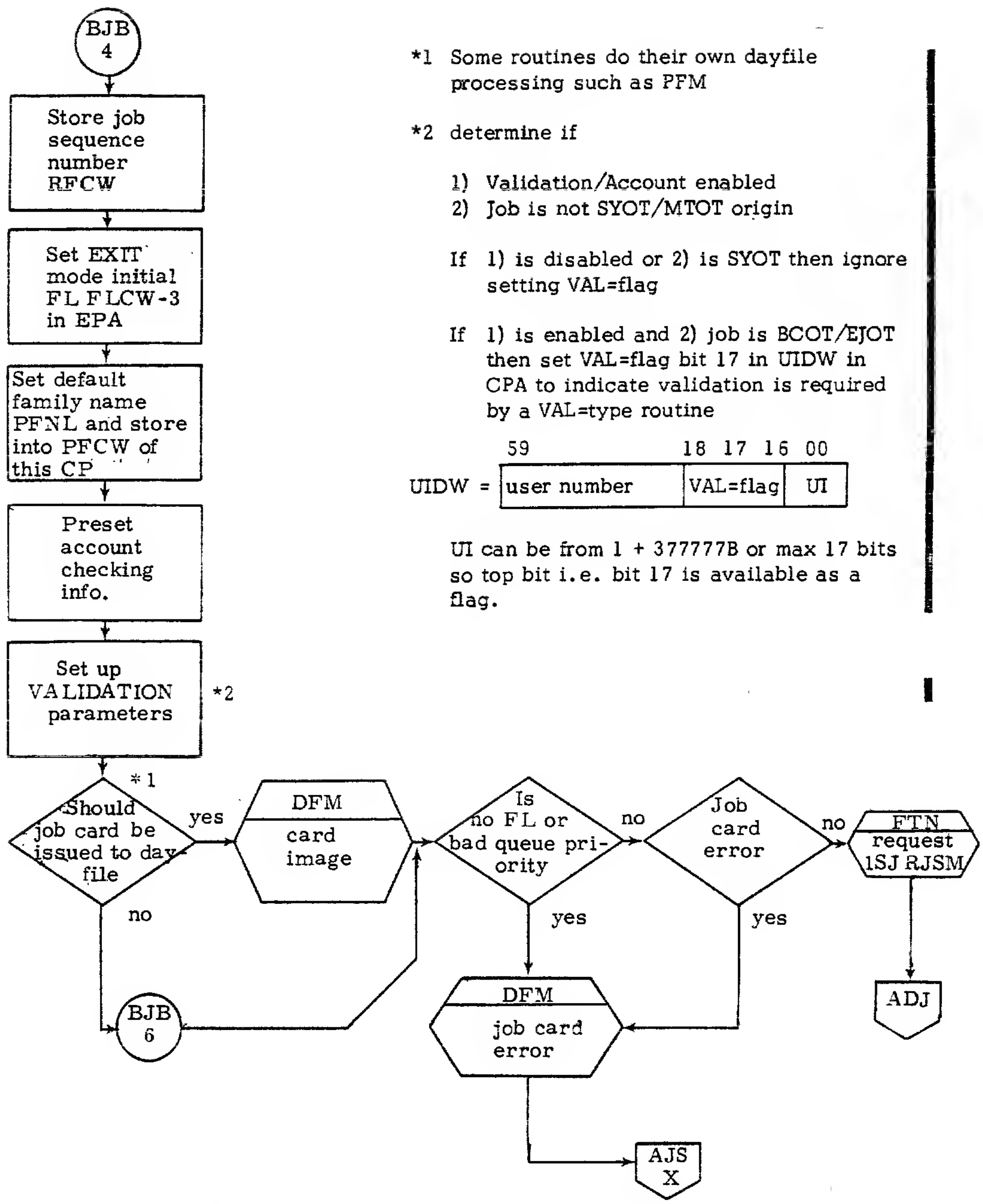
"PROGRAM STOP AT xxxxxx." = The monitor detected a program stop instruction at address xxxxxx.



Processor Table

Begin Job Processors	
SYOT	BBC
BCOT	BBC
EIOT	BBC
MTOT	BMT

Figure 6-8. 3AA — Begin Job



*1 Some routines do their own dayfile processing such as PFM

*2 determine if

1) Validation/Account enabled
2) Job is not SYOT/MTOT origin

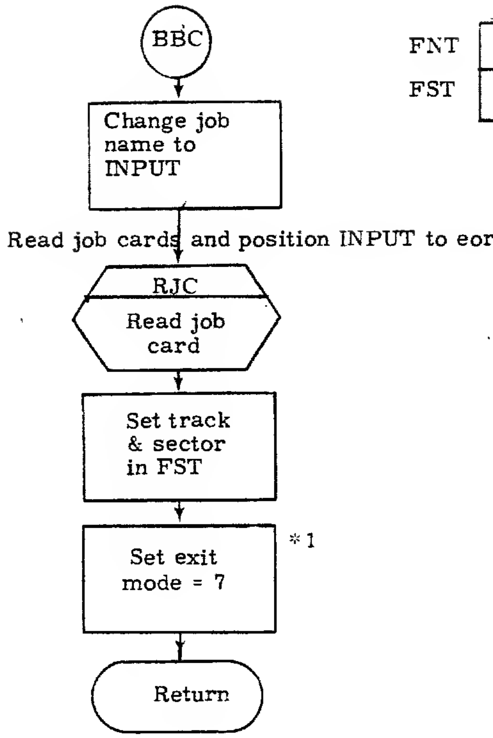
If 1) is disabled or 2) is SYOT then ignore setting VAL=flag

If 1) is enabled and 2) job is BCOT/EJOT then set VAL=flag bit 17 in UIDW in CPA to indicate validation is required by a VAL=type routine

	59	18 17	16 00
UIDW =	user number	VAL=flag	UI

UI can be from 1 + 377777B or max 17 bits so top bit i.e. bit 17 is available as a flag.

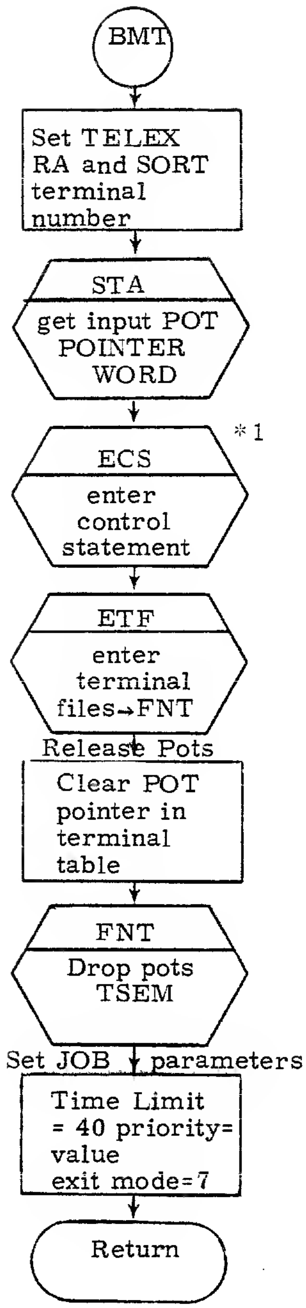
Figure 6-8. 3AA — Begin Job (Continued)



FNT	INPUT		Job origin	INFT	CP number
FST	ID	EST	1st track	Current TRK sector	queue priority

*1 In exchange package

Figure 6-9. 3AA — Begin Batch Job



*1 Read job card from TELEX pot and sets up control statement

Figure 6-10. 3AA — Begin Multi-Terminal Job

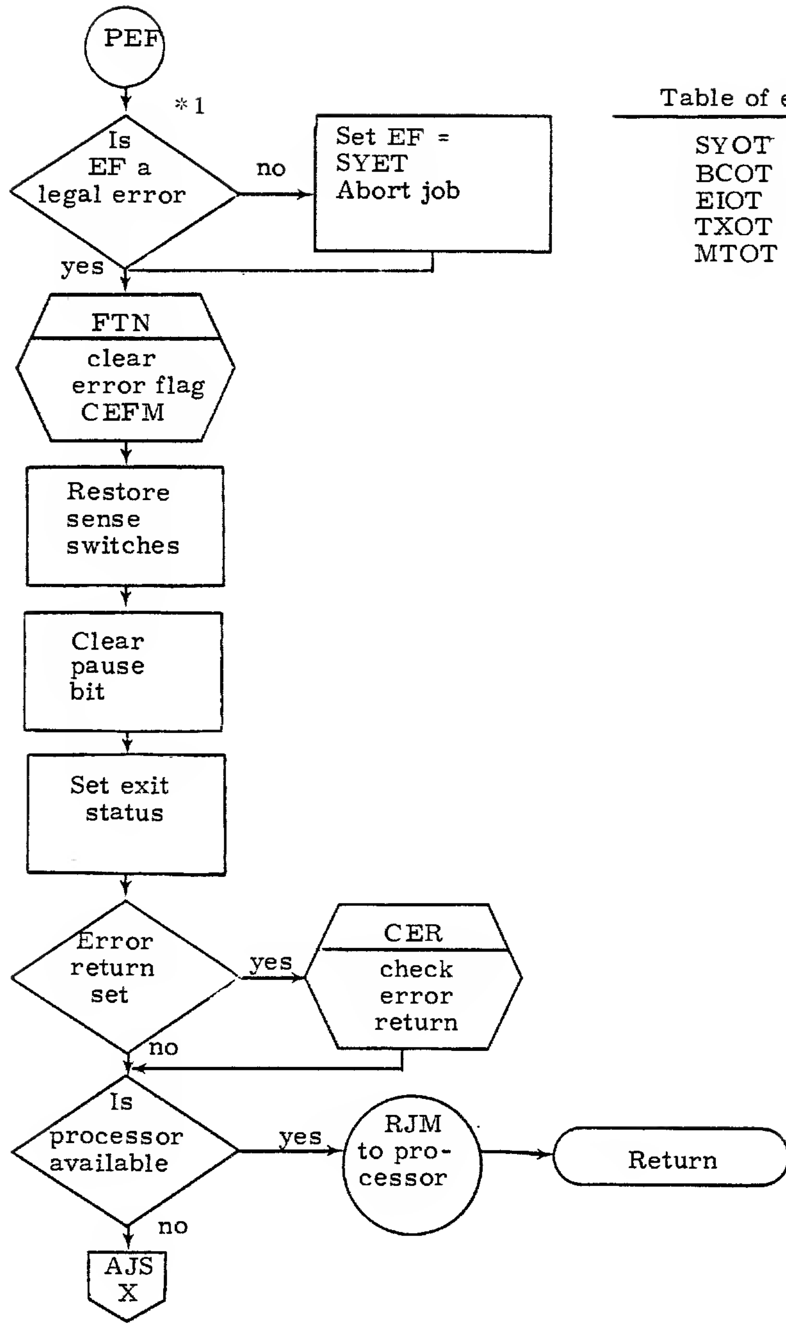


Table of error processors

SYOT	EBC
BCOT	EBC
EIOT	EBC
TXOT	EBC
MTOT	EBC

*1 is EF MXET max. error size

Figure 6-11. 3AB — Process Error Flag

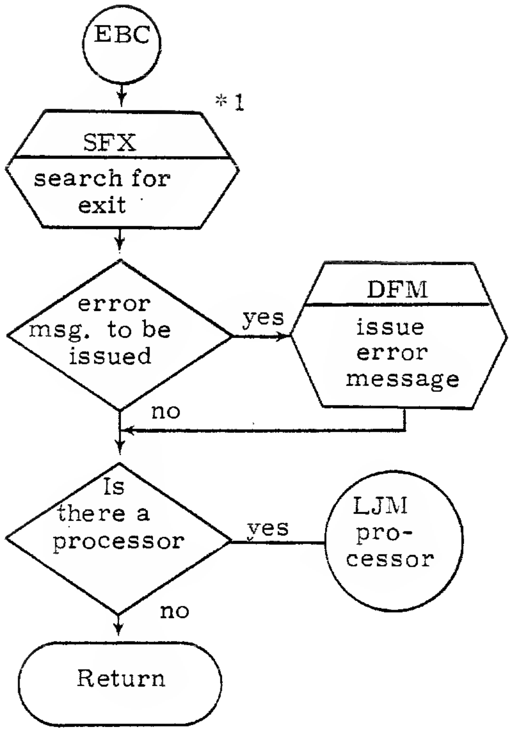


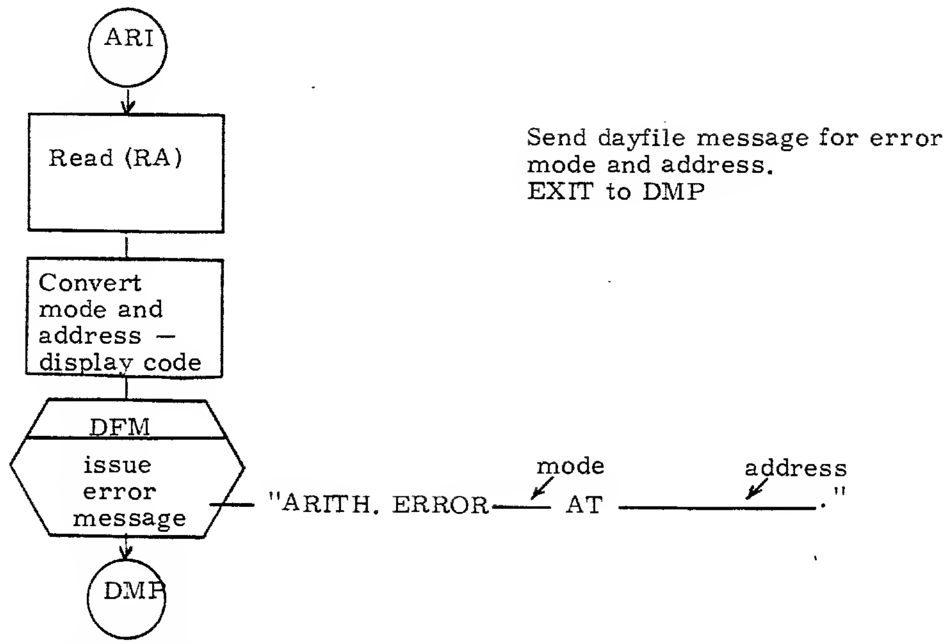
Table of Processors

TLET	TLI	Time limit
ARET	ARI	Arith. error
PCET	MCL	Monitor call error
PSET	PST	Program stop

*1 Look for exit card

Figure 6-11. 3AB — Process Error Flag (Continued)

Process Arithmetic Error



Program Stop Processor

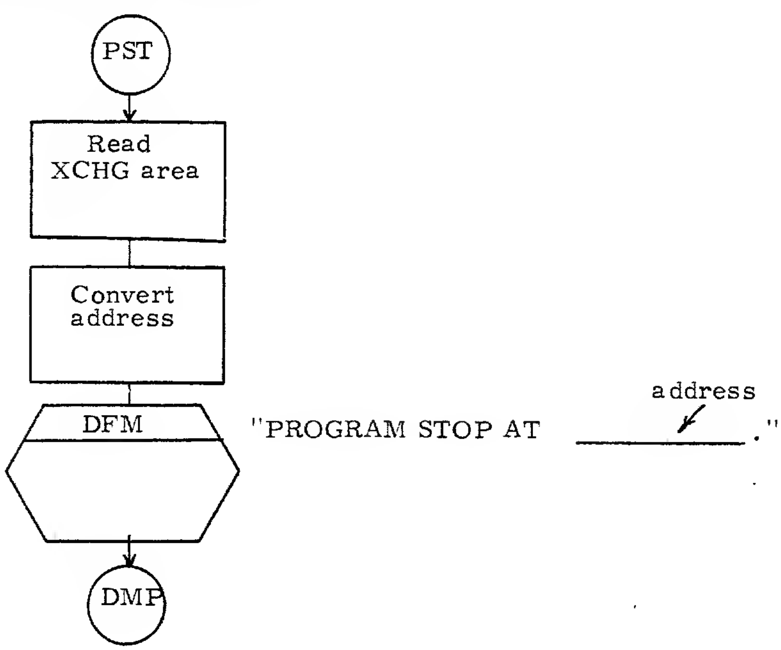
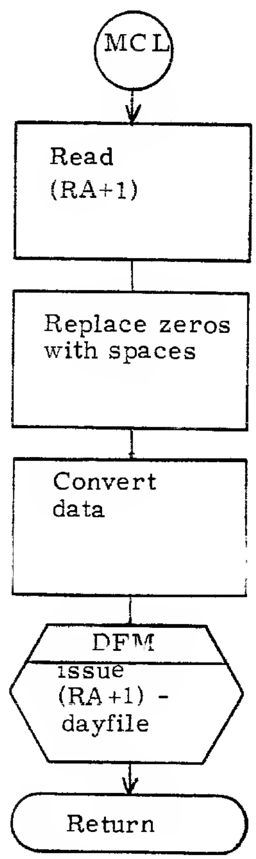
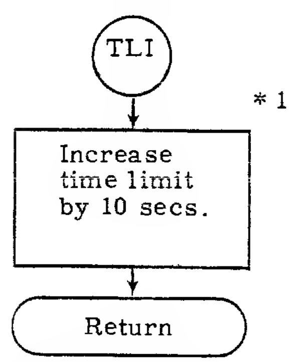


Figure 6-11. 3AB — Process Error Flag (Continued)

Monitor Call Processor



Time Limit Error



* 1   Let user finish error processing if possible

Figure 6-11. 3AB - Process Error Flag (continued)

6.2.3 TCS — Translate Control Statement

TCS translates control statement in the following manner.

1) Read the next statement from:

   a) The control card buffer in the control point area. If necessary, the buffer is reloaded from the "INPUT" file.

   b) The message buffer for "DIS" type programs.

   c) A CM location for an executing program.

2) Programs loaded from the system have their parameters cracked with KRONOS separator equivalences (unless a *SC SYSEDIT directive was used when entering the program into the system).

   Local file program loads have their parameters cracked with SCOPE separator equivalences (unless a "/" prefix character exists on the control card).

   a) KRONOS parameter cracking —

      Delete all spaces imbedded within the statement, up to the character "." or ")". Any character not in the standard FORTRAN set (i.e., $> \leq$; ETC.) are not allowed within the statement. However, they may be used in the comment.

      NOTE: A separator character is one of the set +-/=,($

   b) SCOPE parameter cracking —

      Spaces are treated as separators. All special characters are translated to a 4-bit code.

3) Search the list of special control statement names for a match with the statement being processed. (CTIME OR RTIME)

4) Extract the first 7 or less characters from the statement up to a separator character and search the file name table for a file assigned to this job with this name. If a find is made, the field length will be restored if it is different from the amount set by the last RFL control card, or by the last call to CPM to set running field length. If such a file is found which is on a mass storage device, and it is in absolute code format, the file is read to central memory as a CPU program. If the file does not reside on mass storage, the job is aborted. If the file is in relocatable code format, control is transferred to the relocatable loader. The arguments for the program call are extracted from the control statement and stored in the argument region of central memory, RA+2 - RA+63B. The CPU is requested to begin execution of the program.

5) Search the central library for a program with the name on the control statement. If such a program is found and the program contains an RFL= entry point, the field length will be set accordingly. Otherwise, it will be set as in 4 above. Then, the requested program will be loaded and execution will begin with the arguments stored as in 4 above.

6) If the statement name is a three-character name, with the first character an alpha, it searches the PP library for a program of this name. If found, it places this name with up to two octal arguments as a PP program request and exists to the program. No change will be made in the job field length

   PP program calls via control card are only valid from system origin, or if the caller has system origin privileges and the system is in debug mode.

7) If none of the above are done, the control statement is declared illegal and the job is aborted.

The dayfile messages are:

1) "ILLEGAL CONTROL CARD." = The control statement could not be identified by TCS.
2) "TOO MANY ARGUMENTS." = The number of arguments on the control statement exceeds that allowed by the program.
3) "FORMAT ERROR ON CONTROL CARD." - An error has been detected in the format of the control statement.
4) "PROGRAM FILE EMPTY." = A load of an empty data file was attempted.
5) "COMPILER NOT IN LIBRARY." = An LDC control card requested loading of a compiler not on the system.
6) "LOADER MISSING." = Either CALL or LDR = were not found in the library.
7) "IMPROPER VALIDATION." = A validation program (containing a VAL= entry point - account or charge) is required before continuing.
8) "ADDRESS ERROR." = CM address in call is beyond the FL.

The operator message available is:

"WAITING FOR STORAGE." = Job processing is waiting for memory to be made available.

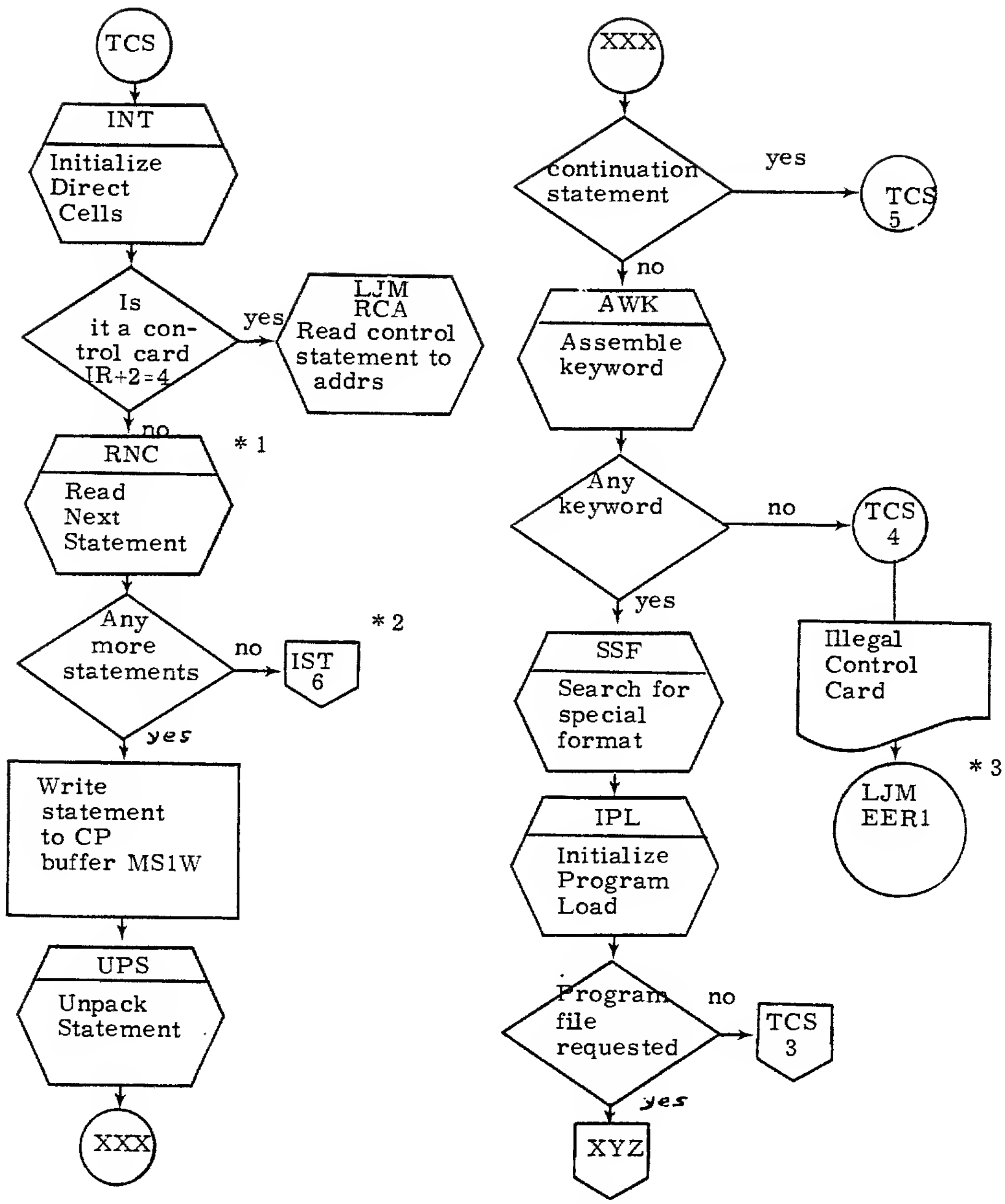
The routines used are:

0BF - Begin file
0DF - Drop special ID files

The direct location assignments are:

name	value	description
PF	65	Program format
CA	66	Character address
KA	67	Keyword start address

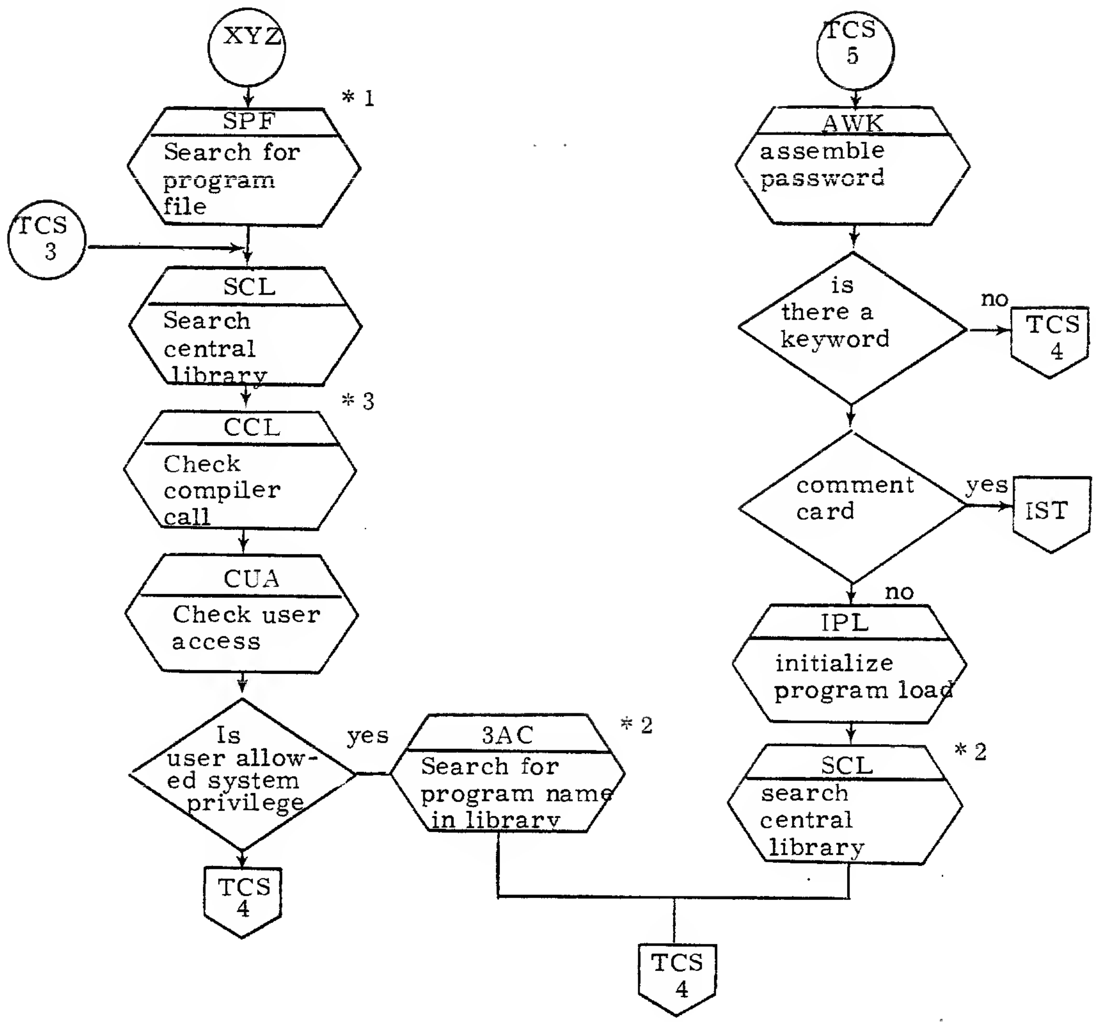
TCS will process the two control cards CTIME and RTIME directly in subroutine SSF - Search for Special Format. TCS gets the CPU time for CTIME or the current time for RTIME and issues the time with the appropriate message to the callers dayfile. Then



* 1 If buffer empty and no eor RNC will read next buffer
* 2 Store statement pointers
* 3 Process error and exit.

Figure 6-12. TCS - Translate Control Statement

Main Loop (continued)



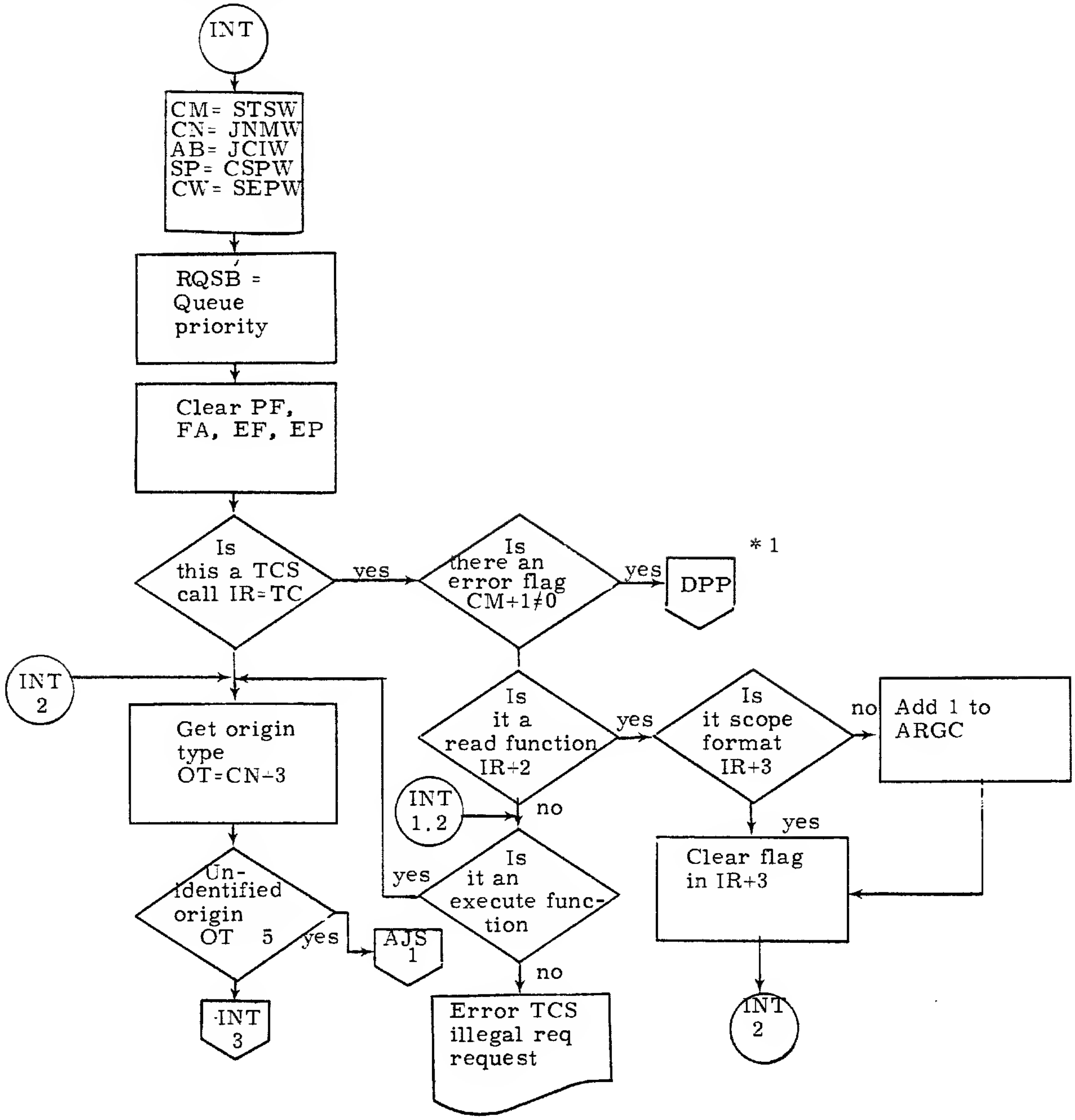
* 1 If prog in local FNT files 1AJ tells LDR to load "LDR=" (relocatable loader) puts CP in "W" status with P="LDR=" entry points and drops from PP.

* 2 Will exit to ILLEGAL CONTROL CARD if name not found. If name is found it will load and execute.

* 3 Return with CP in "W" status. Check for $LDC call from TELEX and load compiler.

Figure 6-12. TCS = Translate Control Statement (Continued)

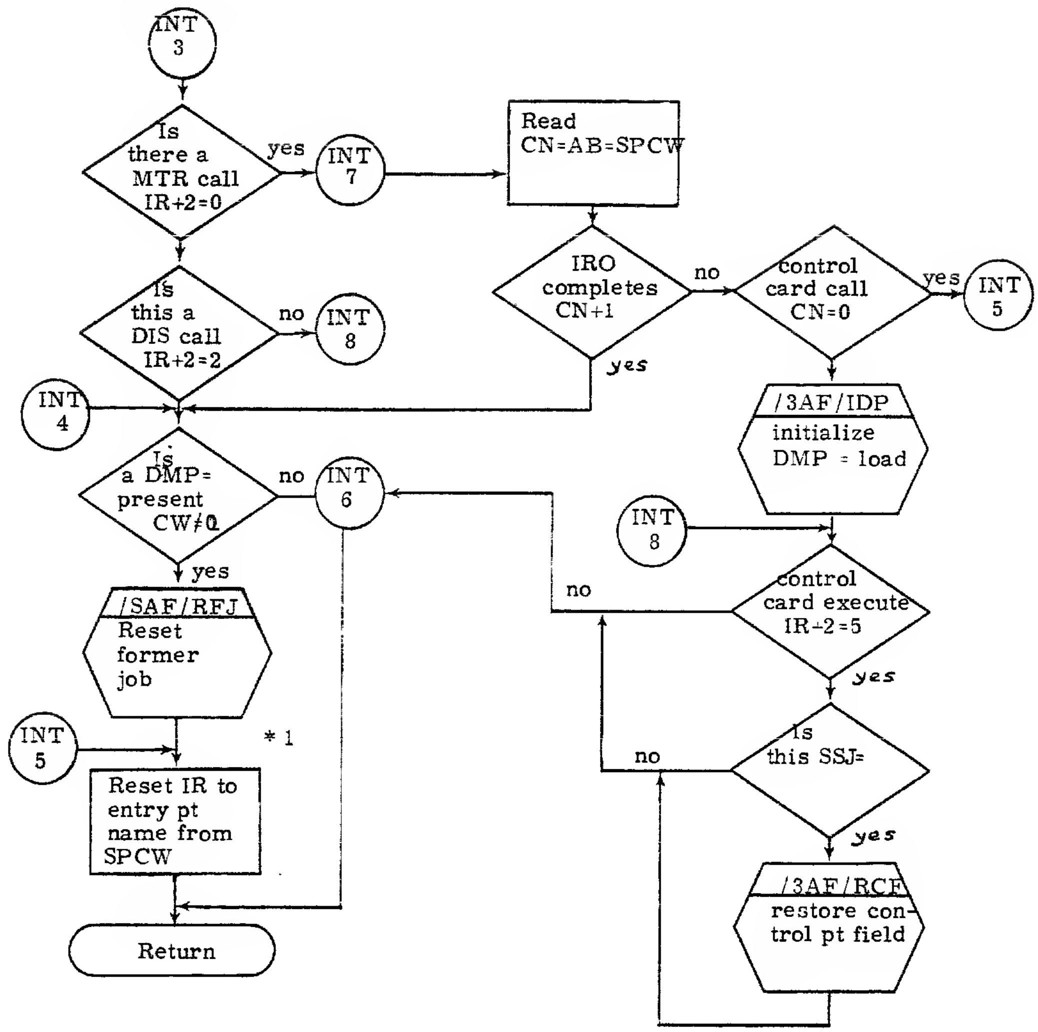
INT - Initialize Direct Cells



* 1 Exit let error processor catch it

Figure 6-12. TCS = Translate Control Statement (Continued)

INT - Initialize Direct Cells (cont'd)



* 1 Will restore the former job from an SSJ=, DMP=, if necessary

Figure 6-12. TCS - Translate Control Statement (Continued)

Issue Statement to Dayfile

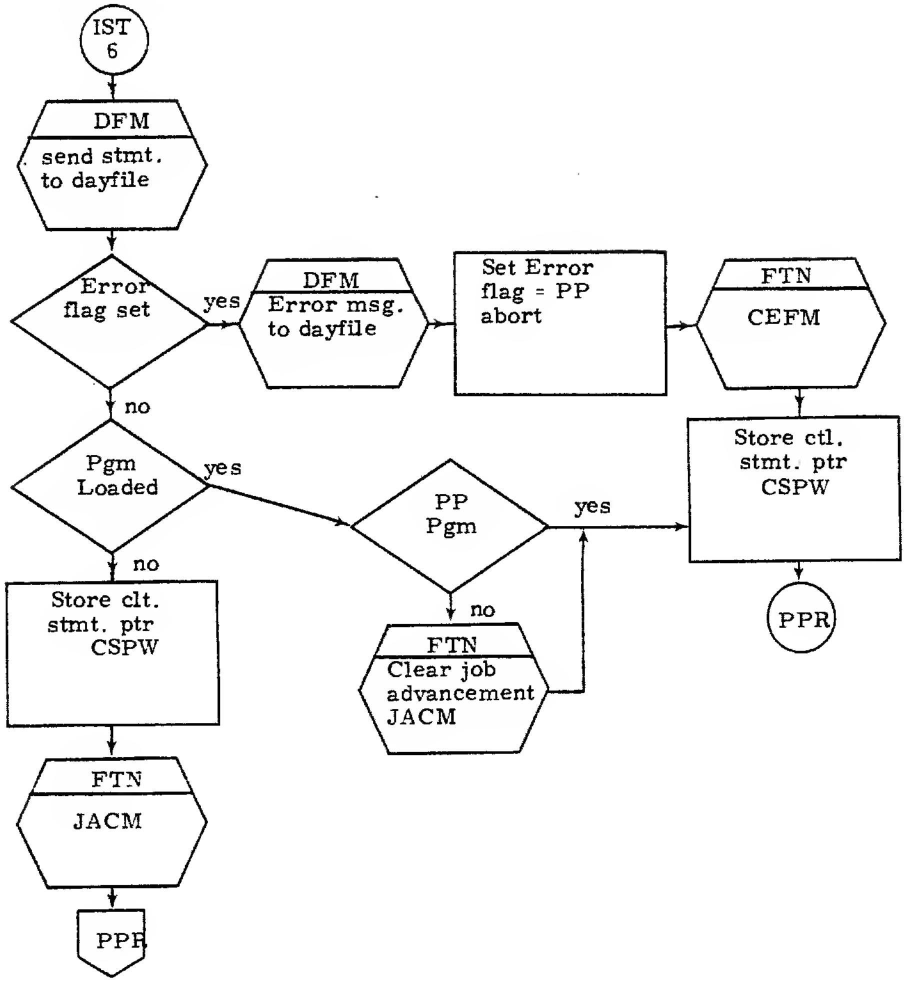


Figure 6-12. TCS - Translate Control Statement (continued)

TCS is ready to advance to the card following CTIME or RTIME. All other control cards will force some routine to be loaded or an abort.

6.2.4 LDR - Overlay Loader

LDR will load absolute overlays in response to CPU program requests. 1AJ can request LDR by return jump to /LDR/CLD which is loaded by *CALL COMPCLD. CLD searches the central library, and, if an entry is found, LDR is called. 1AJ uses this technique for other type calls such as /LDR/LCP load central program. 1AJ effectively uses whatever pieces of LDR it needs to get routines loaded.

LDR calls from the CPU routine are:

	18		1			18
RA+1	LDR	0	R			addr

Where:

R - Auto recall if desired

Addr - Address of request

NOTE

See Section 12 for a detailed description of LDR.

6.2.5 3AC - Search Peripheral Library

3AC searches for the program name in the peripheral library. If the entry is found, it puts the name in its IR and exits to PPR.

The main routine it uses is SPL. The SPL entry is (AB-AB+4)=assembled name, and it exists if it is not found or if it is called to the program. SPL uses IT, CA, and AB-AB+4, and calls AOD and SLT.

6.2.6 3AD - Search For Overlay

3AD performs an end around search of the overlay file for an overlay of the requested level.

Its dayfile messages are:

1) "OVERLAY FILE NOT FOUND" = Requested file is not available.
2) "OVERLAY FILE EMPTY." = No data appears in requested file.
3) "OVERLAY NOT FOUND." = Requested overlay is not on file.
4) "ILLEGAL ACCESS TO EXECUTE ONLY FILE." File is execute only.
5) "FILE NOT ON MASS STORAGE."
6) "ENTRY POINT NOT FOUND." = Requested entry point is not on file.

6.2.7 3AE - Load Copy Routines

3AE contains subroutines used to load programs.

Its dayfile messages are:

1) "OVERLAY NOT FOUND." = Requested overlay was not found.
2) "FL TOO SHORT FOR PROGRAM."
3) "ILLEGAL LOAD ADDRESS." = Load address. LT. 2
4) "UNIDENTIFIED PROGRAM FORMAT." = The file requested to be loaded was not in a recognized format.
5) "ECS LOAD ERROR." = Bad load address from ECS.

6.2.8 3AF - Special Entry Point Processing

3AF contains subroutines for processing DMP= and SSJ= entry points.

A description of the subroutines is as follows:

1) RCF - Restore control point area fields.

   Entry - If no job activity

   Exit - Control point area fields restored. Files with special ID set are dropped.

   Calls - 0DF, SPR

2) IDP - Initialize DMP= program load on RA+1 call

   Entry - If DMP= CP program to be loaded

   Exit - To program loaded

3) PSR - Process special processor request

Entry - (A) = directory address from CLD

Exit - to 1RO for DMP= rollout

4) RFJ - Reset former job

Entry - If DMP= job to be restarted

Exit - to 1RI for DMP= rollin

5) SDP - Start up DMP= job

Entry - Upon return from 1RO at DMP= rollout completion

Exit - None drop PP

6) SPR - Set priorities

Entry - (RCFA - RCFA+4) SSJ= priority values

Exit - priorities in control point are set according to (RCFA-RCFA+4). If (RCFA-RCFA+4) are all zero, no action will occur.

7) TCA - Transfer control point area fields (SSJ=)

Entry - (CSED-CSED+4) Special entry point word

Exit - None, drop PP

6.3. 1CJ - COMPLETE JOB (FINISH UP JOB AND CLEAR CP)

1CJ performs all of the job termination procedures.

These include:

1) Release storage
2) Release assigned equipment
3) Release any common files used by job
4) Dropping of any scratch files used by job
5) Release all output files to output queue
6) Place the accumulated CPU time in the dayfile
7) Append the control point dayfile to the end of the print file, and flush dayfile buffer
8) Updates Resource Files

The 1CJ call is:

1CJ	0	CP	0

CP = control point number

The 1CJ dayfile messages are:

1) "CP xxxxxx.xx SEC." = Accumulated CPU time for the job.

2) "CM xxxxxx.xxx KWH." = Central memory usage expressed as kilo-word-hours. (Field length X time)

3) "MS xxxxxx.xxx KPR." = Mass storage usage expressed as kilo-physical records transferred.

4) "MT xxxxxx.xxx KPR." = Magnetic tape usage expressed as kilo physical records transferred.

1CJ uses the following routines:

0BF - Begin file

0DF - Drop file

0RF - Update resource files

6.4 1RO JOB ROLLOUT ROUTINE

1RO performs job rollout in response to a calling program (such as the job scheduler or the system display) or a dump field length function from 1AJ.

The 1RO call is:

IR=

<table>
<tr><td>18</td><td>1</td><td>5</td><td>12</td><td>6</td><td colspan="5">18</td></tr>
<tr><td rowspan="2">1RO</td><td rowspan="2">0</td><td rowspan="2">CP</td><td rowspan="2">FN</td><td rowspan="2">0</td><td colspan="5">N</td></tr>
<tr><td>0</td><td>C</td><td>F</td><td>U</td><td>FL</td></tr>
</table>

Where:

CP = Control point number

FN = 0 Rollout

= 1 Selective rollout to file DM* according to DMP= parameter

N = Error flag for TXOT job (function 20)

= DMP= parameter (function 1).

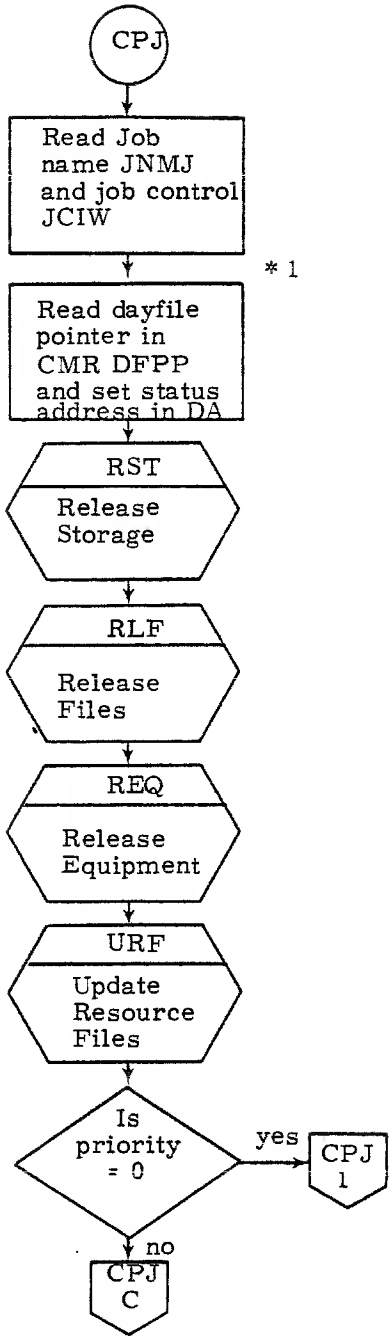
Bit 14 C = Create DM* file only.

Bit 13 F = Dump FNT entries to file DM* .

Bit 12 U = Create DM* as an unlocked file.

Bit 0-11 FL = 0, dump CP area and entrie FL.

≠ 0, dump CP area and FL* 100B.



Release all memory for this control point. RST issues RSTM request zero words of memory.

Close and clear all FNT/FST and drop all unused tracks for this control point (file OUTPUT will be checked later and if it exits then taken care of in RPF)* 2.

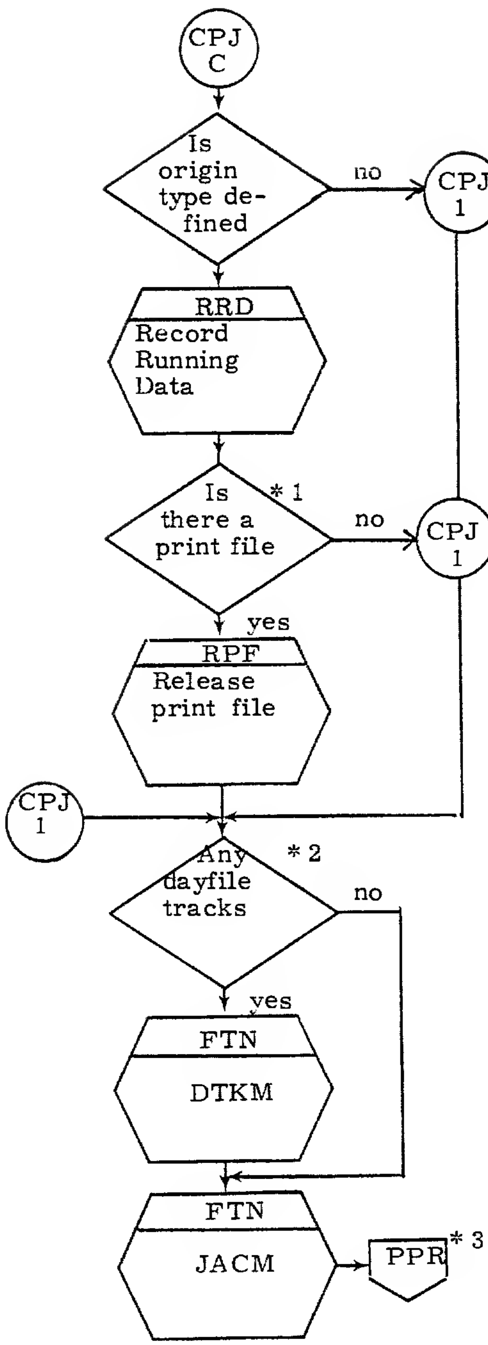
Release all equipment assigned to this control point.

Use 0RF clear all entries in RESEXDF for this control point.

* 1 Used at CPJ1

* 2 See next page

Figure 6-13. 1CJ - Complete Job



Record CPU and PPU, CM, MS, and mag tape usages and issue dayfile messages.

Set print file name to job name, set type to "PRFT" or "PHFT" (Sect. 2) append dayfile to end of file and release file from this control point.

Drop tracks

Set job advancement control and request monitor to set this control point clear.

* 1 Is there a file whose name is OUTPUT?

* 2 Use cell DA set on last page

* 3 JACM will drop the PP when op code =2 or 3 in OR.

Figure 6-13. 1CJ - Complete Job (continued)

1RO uses the 0BF, begin file, routine. Its direct location assignments are:

Name	Value	Description
FS	20-24	FST entry (5 locations)
NT	25	Next track pointer
FW	26	FNT word count or central memory index
SC	27	Sector count terminal output
CN	30-34	CM word buffer (5 locations)
TW	35	Constant 2
DP	36	Dayfile pointer address
OT	37	Origin type
FN	40-44	FNT entry (5 locations)
TN	45	Terminal number
TT	46-47	Terminal table address (2 locations)
FA	57	Address of FST entry
ZR	60-64	CM zero word (5 locations)
TA	65	TELEX RA
OP	66-67	Output pointer (2 locations)

6.5 1RI JOB ROLLIN

1RI performs job rollin response to a calling program, such as the job 1SJ scheduler or the system DSD display.

Its call is:

	18	1	5	12	12	12
IR =	1RI	0	CP	FN	0	FA

Where:

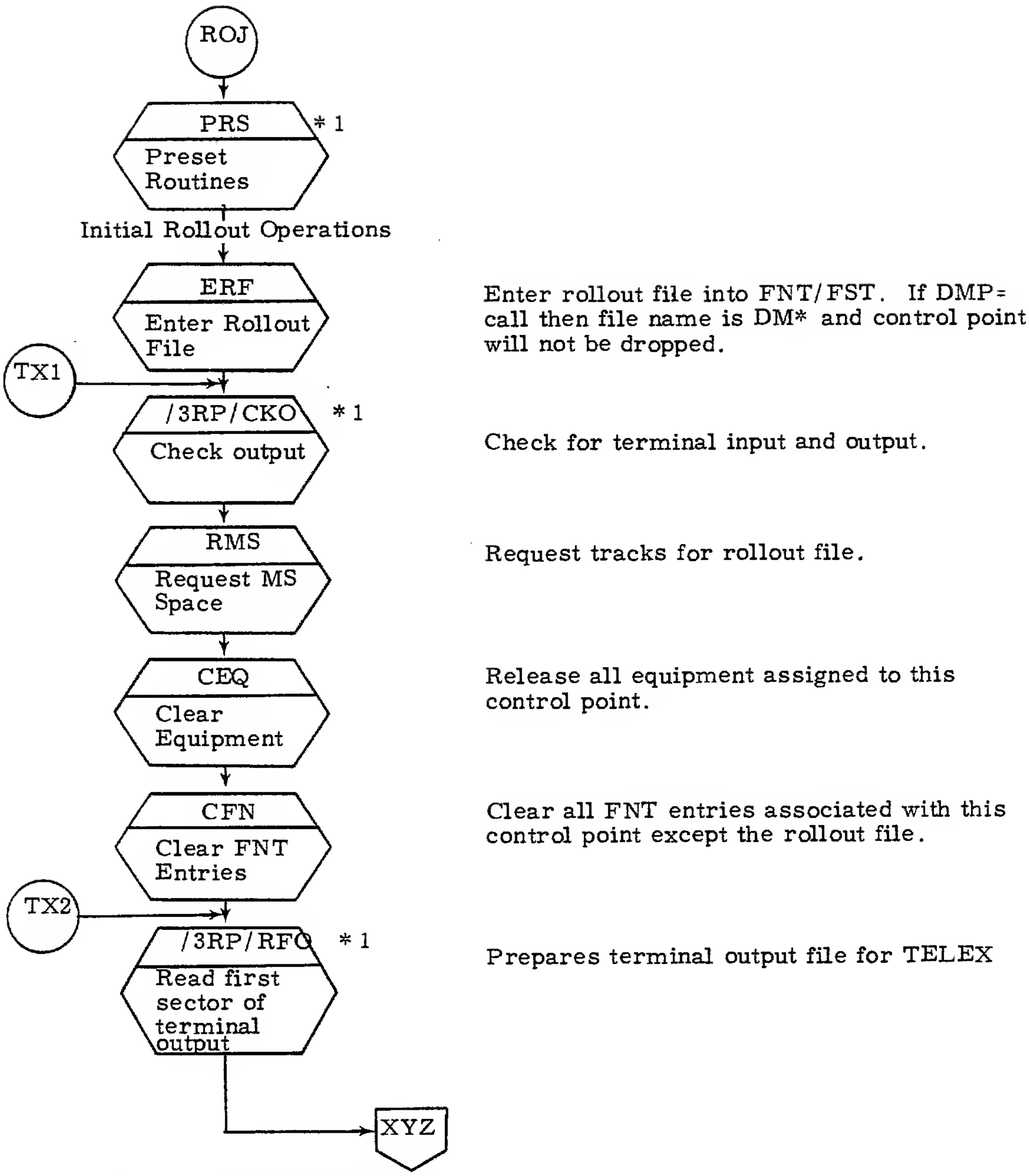
CP = Control point number

FN = 0, Rollin job.

= 1, selective rollin according to special entry point

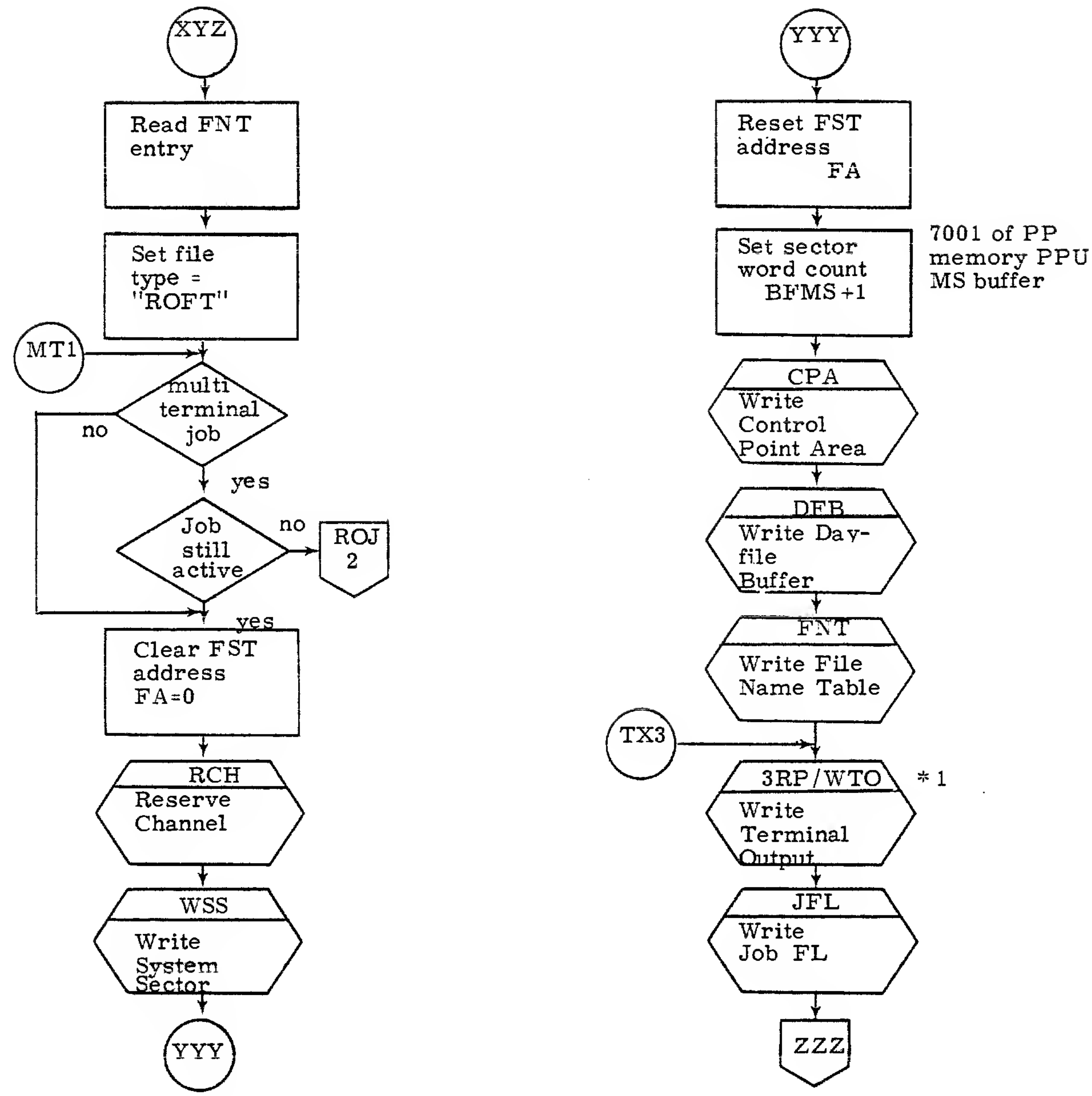
FA = FST address of rollin file

The 1RI dayfile message is "ROLLIN FILE BAD", signifying that an illegal format is detected in the rollin file (see Section 5, paragraph 5.2 for a description of the rollin file).



* 1 Disable all jumps associated with TELEX origin jobs if this is a non-TELEX origin job.

Figure 6-14. 1RO - Rollout Job



* 1 See comment on previous page.

Figure 6-14. 1RO - Rollout Job (continued)

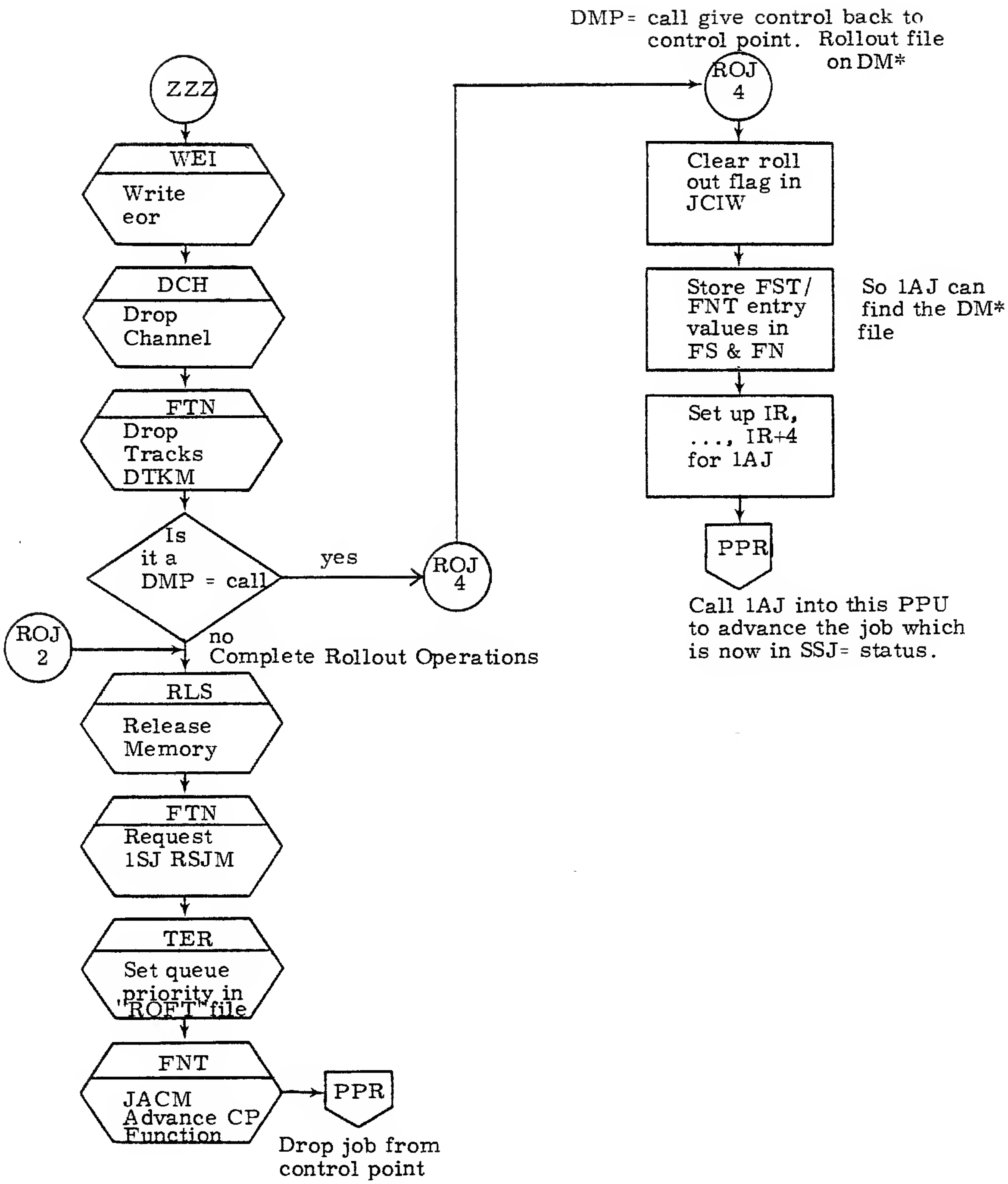
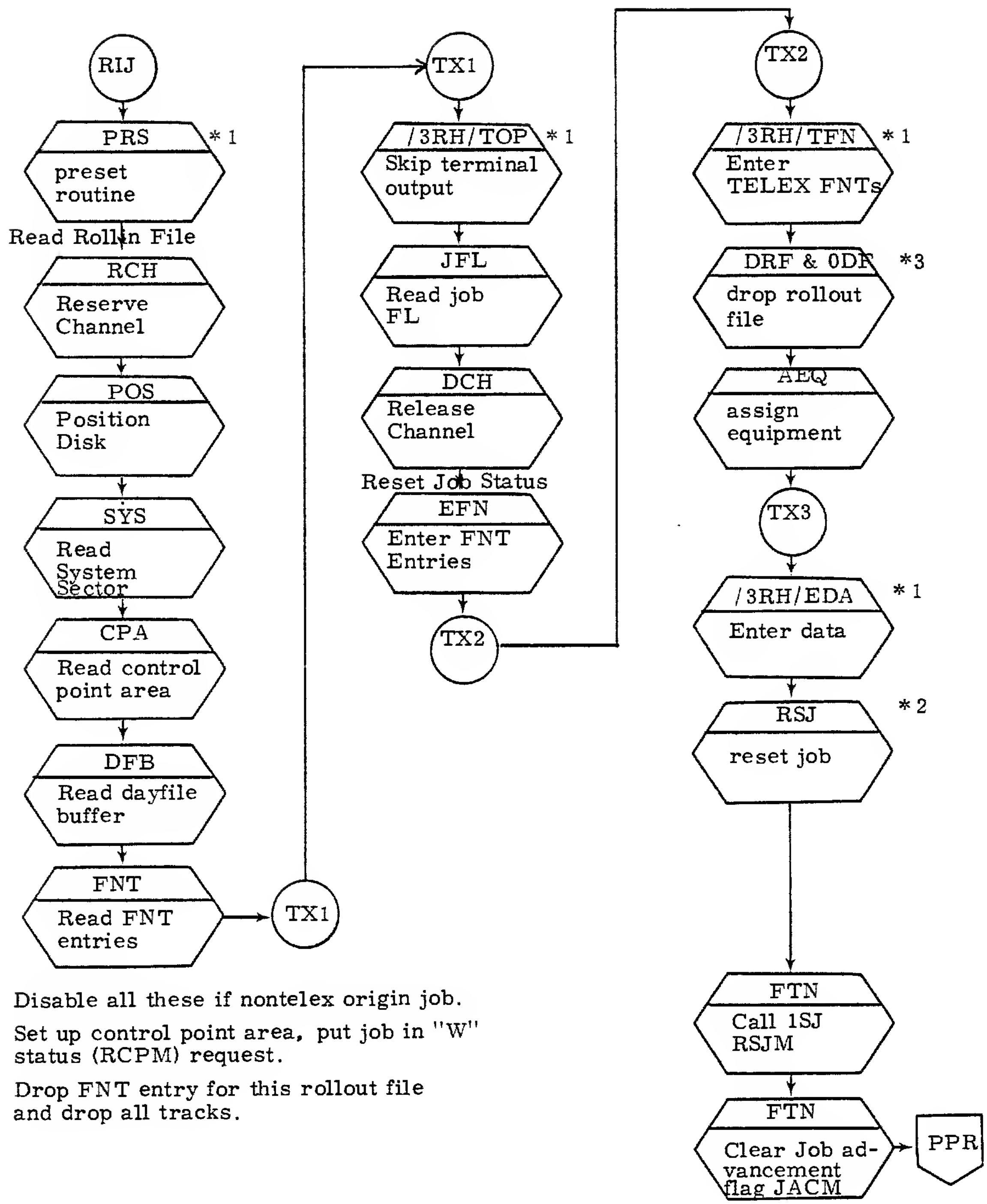


Figure 6-14. 1RO - Rollout Job (continued)

1RI uses the 0RF and 0DF, drop file from rollout file, routine, and has the following location assignments:

Name	Value	Description
FS	20-24	FST entry (5 locations)
DP	25	Address of dayfile buffer pointer
EP	25	Entry point
FI	26	FNT buffer index
CI	27	Central memory index
CN	30-34	CM word buffer (5 locations)
PR	35	Queue priority
TW	36	Constant 2
OT	37	Origin type
TN	40	Terminal Number
TT	41-42	Terminal table address (2 locations)
PP	43	POT pointer
PA	44-45	POT address (2 locations)
TA	46	RA of TELEX
TI	47	TELEX FNT buffer index
FA	57	Address of FST entry
ZR	60-64	CM zero word (5 locations)
EF	65	Error flag hold

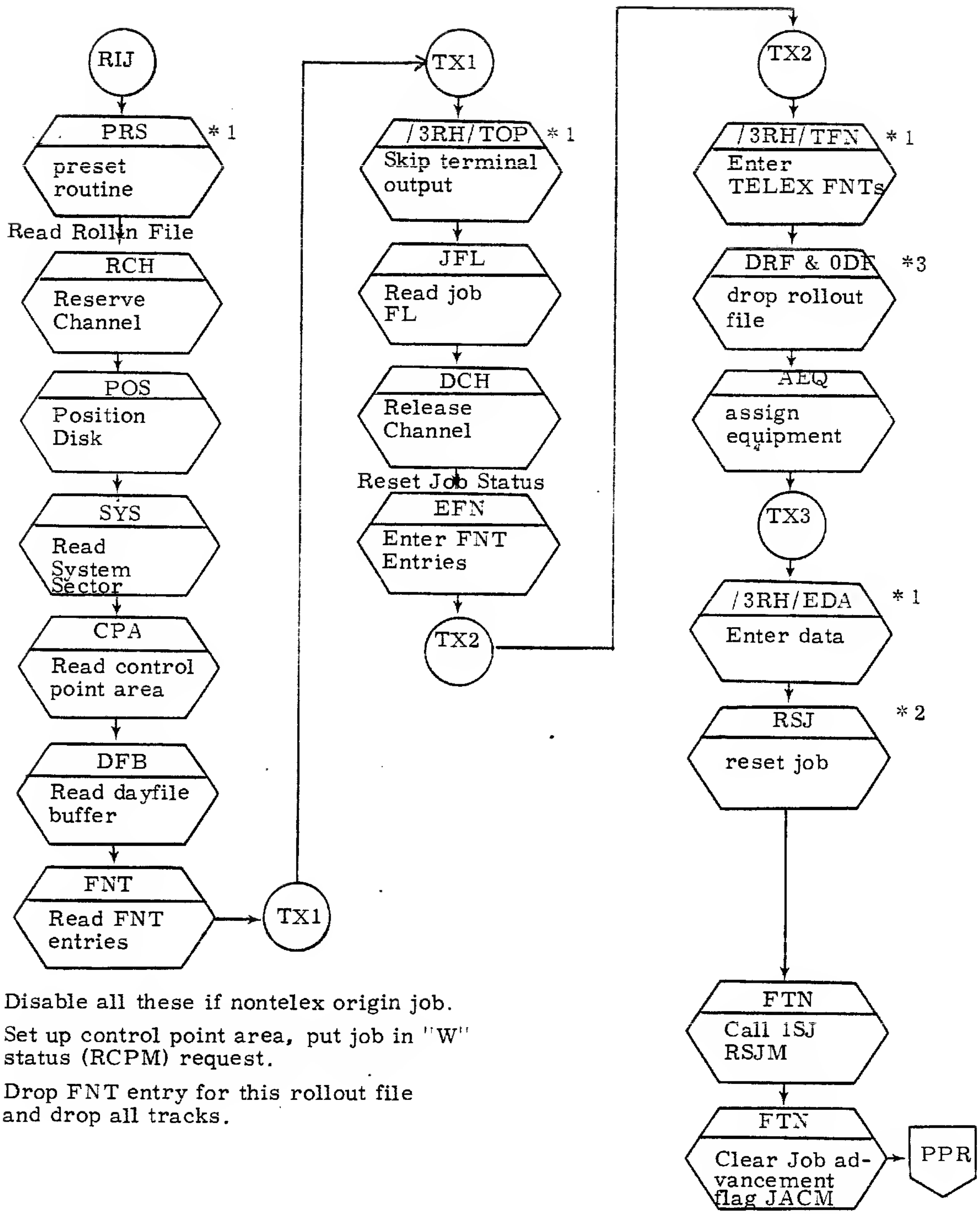


* 1 Disable all these if nontelex origin job.

* 2 Set up control point area, put job in "W" status (RCPM) request.

* 3 Drop FNT entry for this rollout file and drop all tracks.

Figure 6-15. 1RI - Rollin Job



* 1 Disable all these if nontelex origin job.

* 2 Set up control point area, put job in "W" status (RCPM) request.

* 3 Drop FNT entry for this rollout file and drop all tracks.

Figure 6-15. 1RI - Rollin Job

SYSTEM I/O (MASS STORAGE) 7

7.0 INTRODUCTION

All active files residing on Rotating Mass Storage (RMS) are described by a File Environment Table (FET) and by a File Name Table (FNT). The FET is supplied by the user and resides within the job field length. The FET is described in Section 8. The FNT is supplied by the system and is used by system routines to coordinate user requests for I/O and file positioning. Only two other mass storage tables are involved with controlling I/O. These are the Equipment Status Table (EST) and the Mass Storage Table (MST).

7.1 TABLE LINKAGE

The linkage between these tables is simple and reduces system overhead to a minimum. Table linkage is: FNT → EST → MST. The FNT entry for a file consists of two CM words (Figure 7-1). The first word is the FNT word and contains the logical file name for the file. The second word is the File Status Table (FST) word and contains the file status, position, and equipment. The EST entry is one word which describes the device type, the channel(s), and a pointer to the MST entry for this device. The MST contains a complete description of the RMS device showing which tracks are in use and which are available. A detailed description of these tables, all of which reside in CMR, is available in Section 2.

7.2 MASS STORAGE TABLE

MST entry can be thought of as a Track Reservation Table (TRT) with a 20B word header. The header words describe the TRT, as well as provide other pertinent system information describing the device. The TRT provides information about each track available on the RMS device. Since TRT sizes vary depending on the device type, the MST entries vary in size accordingly. However, each MST entry begins on a 10B word boundary so that they can be addressed with the 12-bit field in byte 4 of the EST entry. The MST entries are built at deadstart time by a routine named SET. Permanent file information is taken from the LABEL track by routine RMS. The lengths of the TRT's are outlined in Table 7-1.

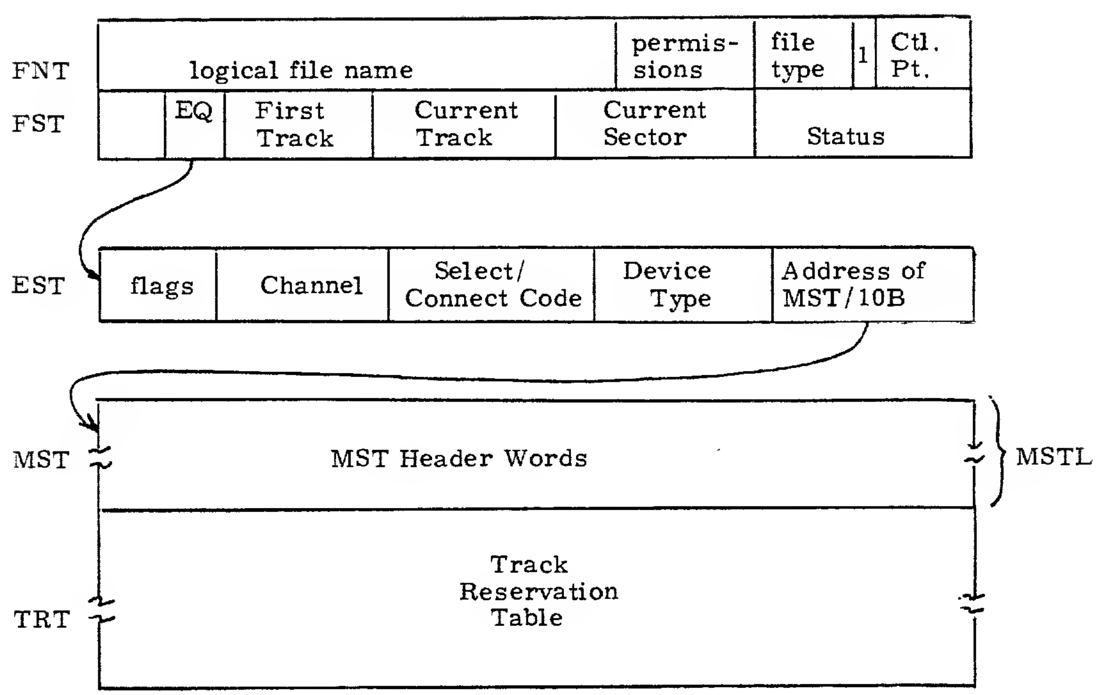


Figure 7-1. RMS Table Linkage

TABLE 7-1. TRT LENGTHS

Device	Mnemonic	in CM Words	Track Count	Logical Sectors/track	Sector buffer size
6603	DA	1000	4000	two zones	502
6638	DB	1000	4000	61B	503
863	DC	100	400		all others
853/854	DD	144	620		502
ECS	DE	dependent on ECS length			
813/814	DF	1000	4000		
821	DH	1000	4000		
844	DI	624	3120	153B	
DDP/ECS	DP	dependent on ECS length			
841	MD	620	3100	100B	
		(above values are octal)			

The TRT lengths above do not include the 20-word header.

The TRT contains single-word entries that define up to four tracks, a link to another track, and control information. Bytes 0, 1, 2, and 3 represent a particular track, while byte 4 contains three 4-bit control settings as follows:

Byte 0	Byte 1	Byte 2	Byte 3	d	w	i

where,

Bytes 0-3 of a given TRT word represent a particular track.

d - A bit is set corresponding to bytes 0-3 to identify the <u>first</u> track of a permanent file chain.

w - A bit setting establishes an interlock of a track. Used by PFM.*1

i - A bit setting for track reservation used by CPUMTR.

From left to right, the three 4-bit control settings correspond to bytes 0 through 3, respectively. This is shown in Figure 7-2.

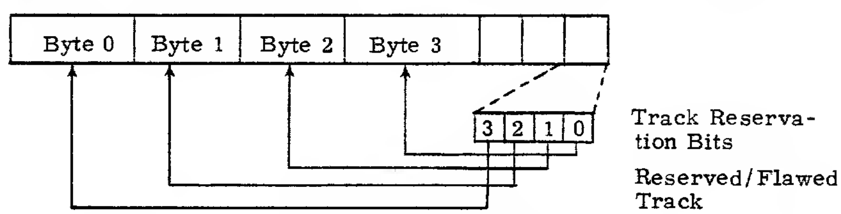


Figure 7-2. Bit Settings for Track Link Bytes

*1 Both IPF and DPF's are interlocked as follows: For PFDUMP, the tracks are interlocked one at a time as they are dumped via the TRT w bits. For PFLOAD, the PF devices are interlocked one at a time as they are loaded via the device unavailable for access bit in the EST.

The track link bytes either contain a pointer to the next track in the chain or indicate the end-of-information sector of a file. These two formats are shown in Figure 7-3.

1	Track Link

0	EOI Sector

Notice that the upper bit marks the difference between track link and EOI sector.

Figure 7-3. Track Link Byte Format

7.3 FILE LINKAGE

Similar to the first and current track fields in the FST, the track link byte contains a number which can be broken down to determine the word within the TRT and the byte within that word which is used to represent the track number. That is, the general link byte format which follows:

11	10	9	8	7	6	5	4	3	2	1	0
Z	X	X	X	X	X	X	X	X	X	Y	Y

where,

Z = 1 for next link in chain in bits 0-10.
Z = 0 for EOI sector number in bits 0-10
X - TRT word relative to word 0 of this TRT
Y - byte within word X.

Figure 7-4 is an example showing file linkage from FST to EST to MST. Notice that the file occupies space on tracks 5, 12, 14, 15, 16, 17 and 20. The EOI is sector 7 of track 20. The EST entry shows that the device is a 6638 so that MST entry is 1020B words long. Also, the FST entry shows that the file is currently positioned at End-Of-Information (EOI). TRT linkage can also go backwards (4012→4002→4007, etc.).

7.4 DISK SECTOR

Every sector, as seen from the user, contains up to 64 CM words (100B). However, the system always prefixes the sector with two header bytes (24 bits). These two header bytes contain file linkage and other information. The general format of a disk sector is shown in Figure 7-5.

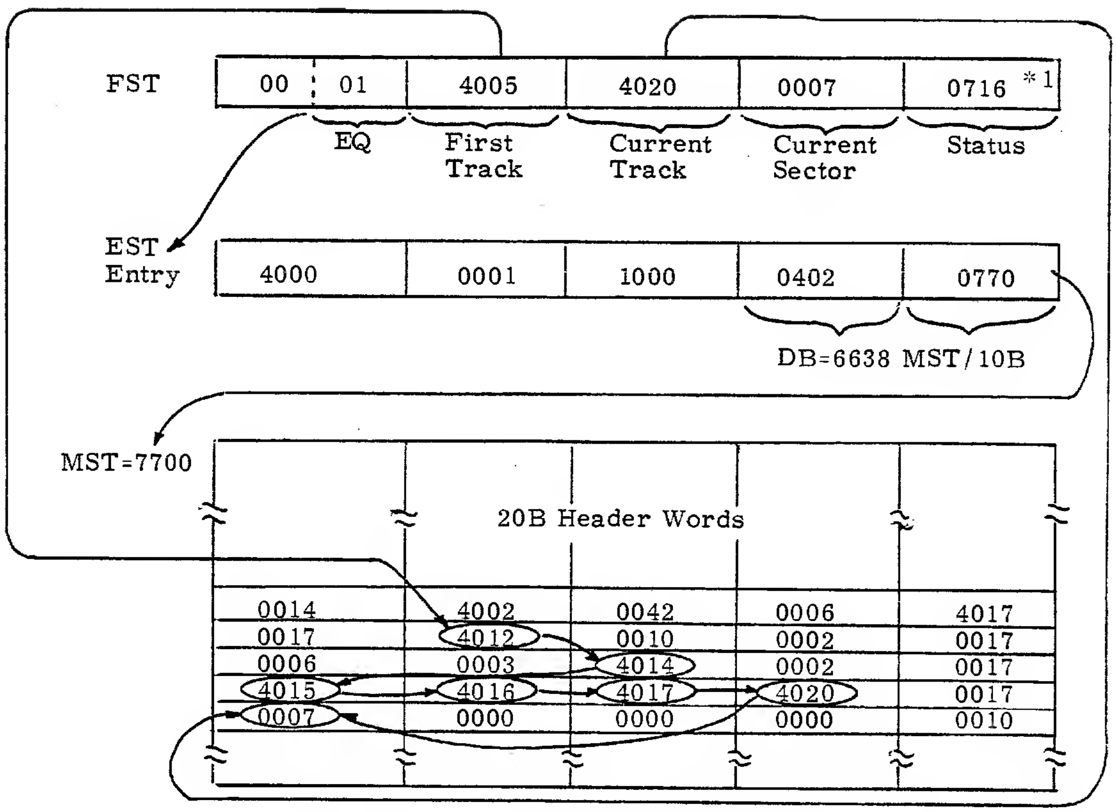
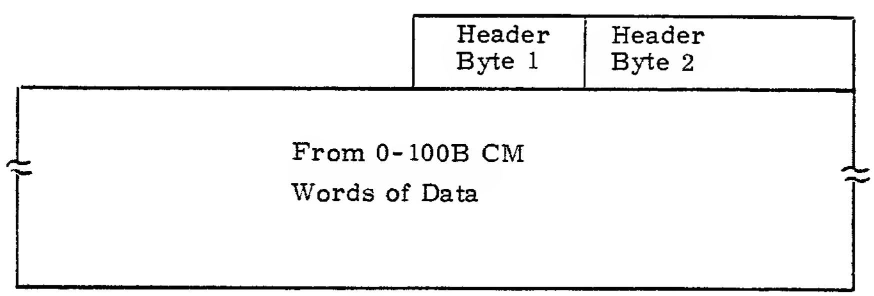


Figure 7-4. Example of File Linkage



*1 Refer to Section 2 (page 2-24).

Figure 7-5. Disk Sector Structure

There are four types of sectors known to the system and marked via the header bytes These are:

- EOR - end-of-record sector
- EOF - end-of-file sector
- EOI - end-of-information sector
- S.S. - system sector

Header byte 2 contains a word count of the number of CM words within the sector as written by the user. The Word Count (WC) is in the range 0 to 100B. If the word count equals 100B, the sector is full. If the word count is less than 100B, the sector is called a short PRU and indicates an End-Of-Record (EOR). Table 7-2 shows the relationships between the various sector types and the contents of the header bytes.

TABLE 7-2. SECTOR HEADER BYTE CONTENTS

Sector Type	Header Byte 1	Header Byte 2	Comment
EOR	Next Sector/Track	$0 \leq WC < 100B$ (PRU)	may or may not contain data
EOF	0	Next Sector/Track	no data
EOI	0	0	no data
S.S.	3777B	77B	system data only
F.S.	Next Sector/Track	WC = 100B	full sector

In Table 7-2, F.S. represents a full data sector and differs from an EOR sector by WC=100 rather than WC < 100 as for the EOR sector.

To differentiate between next sector and next track in header byte 1, bit 2^{11} is set. That is, bit 2^{11} is set to indicate a link to another track rather than a link to the next sector.

The PP common decks that read/write mass storage perform the reading and writing of the header bytes. Also, CIO reads/writes the header bytes for disk I/O. Finally, macros READCW and WRITECW are available to read and write mass storage and magnetic tape files using control words (i.e., header bytes). The PRU format for READCW and WRITECW is in the KRONOS 2.1 Reference Manual.

Again, in Table 7-2, the system sector (S.S.) for a file is indicated by special header byte values. This is done to prevent accidental reading through the system sector itself. SS is always sector 0 of the first track of a file.

Examples of the various sector types are shown in Figure 7-6. The device is assumed to be a 6638; therefore, the sector count is from 0 to 60B. Two situations not shown in Figure 7-6 are an EOR and an EOF as the last sector on a track which link to the next track.

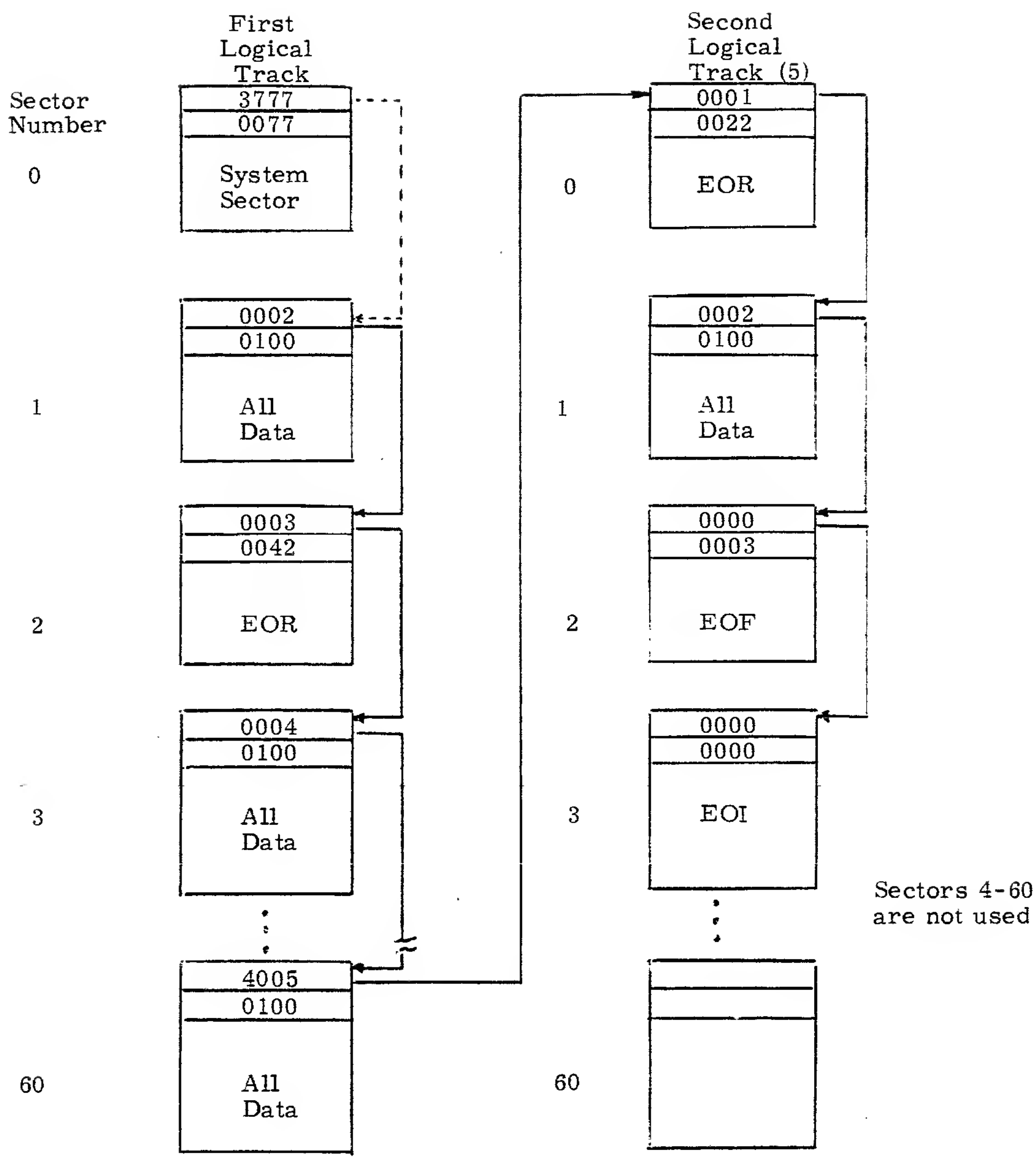


Figure 7-6. RMS File Structure

7.5 PP ROUTINES

Any PP routine requiring disk I/O performs READ/WRITE operations according to the flowchart in Figure 7-7.

A PP routine which writes the disk should begin by issuing monitor function AMSM. AMSM is used to allow MTR to select the best device on which to perform the write. Monitor's device selection criteria are:

1. Unrestricted device with channel free
2. Unrestricted device with channel busy
3. Restricted device with most space

If a system device is required, the monitor function RSYM should be used instead of AMSM. Referring to Figure 7-7, monitor function RTCM provides a track chain for the requesting PP routine. The PP specifies N sectors and monitor returns the first track of a chain of tracks. If the PP routine exhausts the N sectors, additional RTCM functions will be required.

The PP Resident routine SMS determines which driver is currently loaded, and if a different driver needs to be loaded, calls PLL. SMS jumps to MSD+3 after the driver is loaded to perform preset operations. After returning from SMS, the PP routine can use the three driver entry points:

POS - position disk
RDS - read sector
WDS - write sector

These entry points are entered via the RJM instruction. After a write operation, the PP program must issue the monitor function DTKM to drop any remaining tracks and set the EOI sector in the TRT. (This is not done, however, for rewrite-in-place.) The DTKM function is described in Section 3 but is reviewed here. The PP sets up its output register as follows:

CM	CM+1	CM+2	CM+3	CM+4
DTKM	EST Ordinal	First Track	Last Sector	not used

where,

First Track = the last (current) track written.
Bit 2^{11} of this byte is set to 0 so that CPUMTR drops all tracks after this track and stores "last sector" value in track byte in TRT. If bit $2^{11}=1$, "last sector" value is ignored and all subsequent tracks are dropped by CPUMTR.

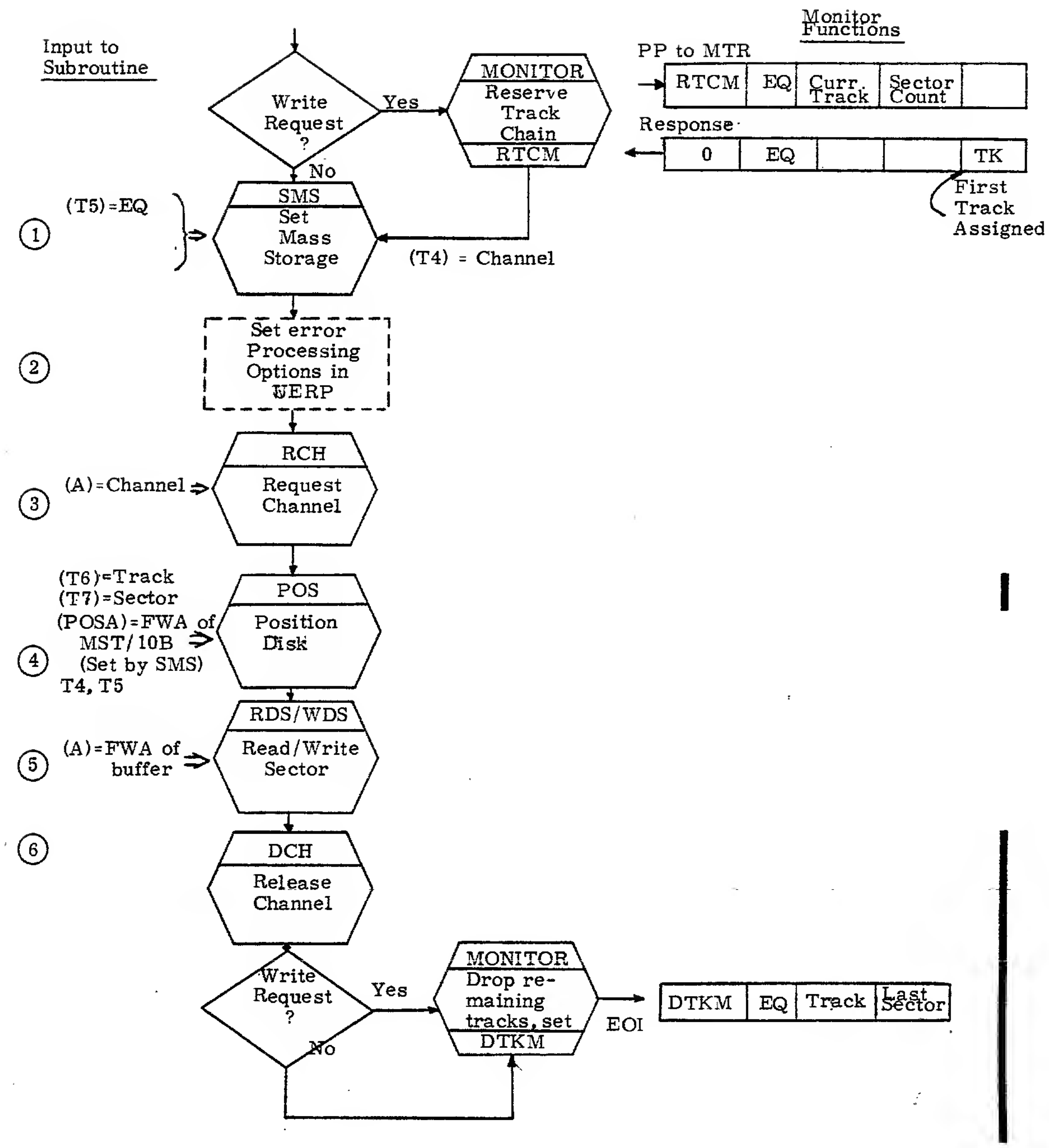


Figure 7-7. Disk I/O from PP Routine

The values above are picked up from direct cells being used as input values to WDS. That is, the contents of T5 can be stored in CM+1, since T5 contains the EST ordinal. The contents of T6 are stored in CM+2 and the contents of T7 are stored in CM+3.

A new monitor function was added to KRONOS 2.1 called SCHM. With this function, a PP routine can request that monitor select a channel for a device when more than one channel is available. This monitor function is used by certain disk drivers to support dual access devices. The drivers that use the function are:

6DH -821 driver
6DI - 844 driver
6DP DDP/ECS driver
6MD- 841 driver

Although these drivers get a channel selected by monitor, it is still necessary for the calling PP routine to reserve the channel with the RCHM monitor function. That is, the SCHM function does not reserve the channel. Along with this feature, the driver (or MTR) issues a release function to the controller after I/O is completed so that the unit is available to another PP on another channel. In this manner, KRONOS 2.1 supports the dual access configuration shown in Figure 7-8.

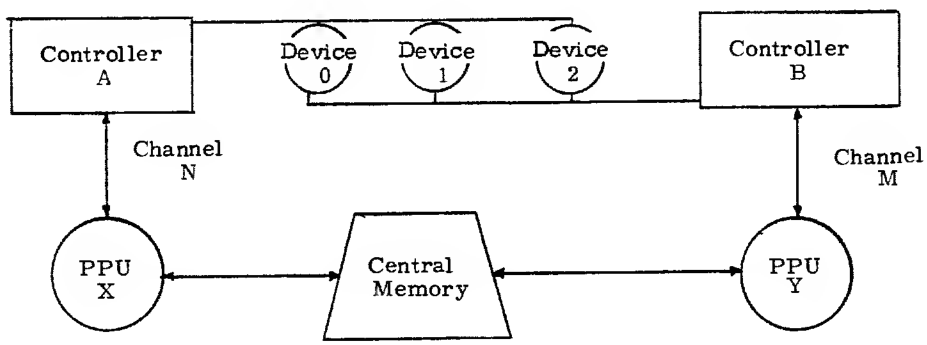


Figure 7-8. Dual Access Configuration (Single Mainframe)

Besides dual access support, much has been added in the area of error detection and correction. Of importance to the system analyst is the error processing options now available with PP programming. Any PP program can select one of the following options when performing disk I/O:

- return control on read error (2^0)
- return control on write error (2^1)
- return control on not ready or reserved status (2^{11})

These options are selected by setting bits 2^0, 2^1, or 2^{11} respectively in PP Resident at the location named UERP. SMS initializes UERP to zero and the program can select the appropriate flag immediately after returning from SMS. For selecting "return on read error", the program can contain the following instruction:

```
AOM    /MSP/UERP
```

When reading, data within the sector is validated based on the following criteria:

1) The word count (WC) in the header byte is less than or equal to 100B.
2) The next sector link in the other header byte is valid.

A return from RDS or WDS with error status is indicated by (A)< 0. If the PP common deck, COMPRNS , is used to read sectors, a return jump to MSR is made when a read error is encountered. The subroutine MSR is supplied by the programmer and should process the read error. To indicate that MSR is available to COMPRNS, it is necessary to define MSR$≠0 during the assembly of the PP program.

Flowcharts from 6DB, the 6638 Disk Driver are shown in Figure 7-10 through 7-14. PRS is entered from SMS while the other three routines POS., RDS. and WDS. are entered via RJM instructions to POS, RDS and WDS, respectively.

All disk drivers are originated at location 600B for loading into PP Resident. The first location (600) contains the device type in display code for driver identification. That is, SMS can ~~determine whether or not a new driver must be loaded. The next two locations~~ (601, 602) contain sector limit values. For example, both cells contain 61B and 153B for 6DB and 6DI, respectively. The next location 603 contains an entry point to the PRESET subroutine within the driver. This is used by SMS only via the following instruction:

```
ZJN   MSD+3   DRIVER ALREADY LOADED
```

Following this are the three entry points:

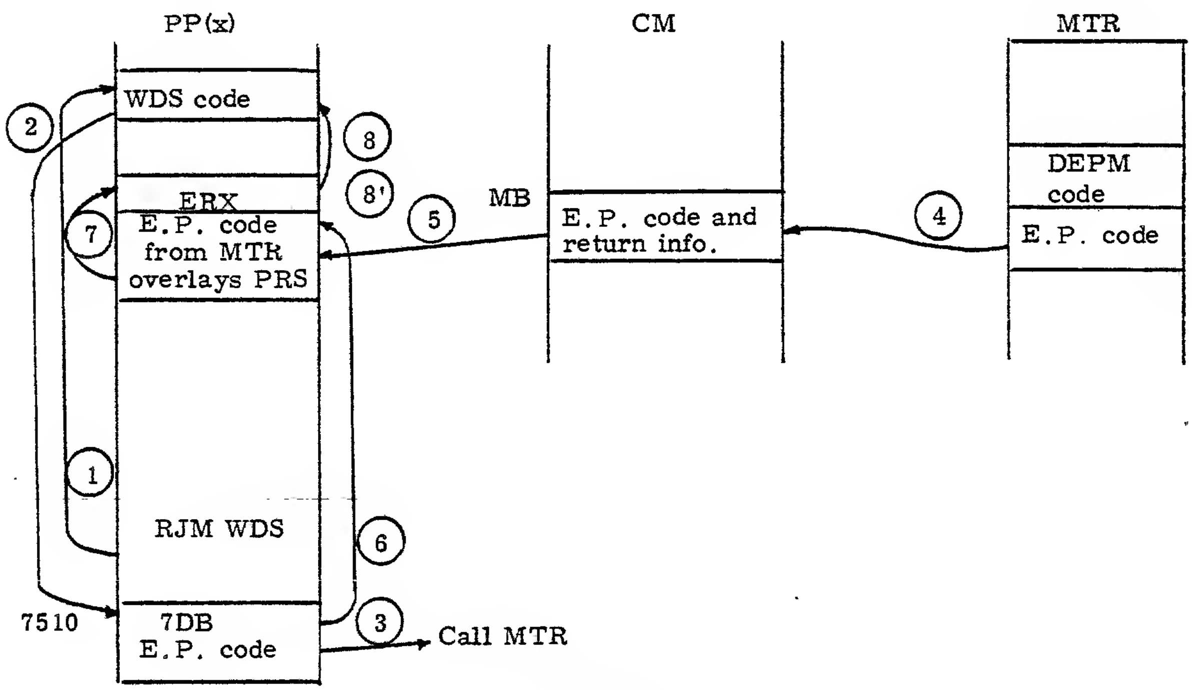
```
606  POS - Position Disk
612  WDS - Write Sector
616  RDS - Read Sector
```

The symbols POS, WDS, and RDS are defined in PPCOM and are the same for all drivers.

547	SMS
600	Driver Name
601	Sector Limit Zone 1
602	Sector Limit Zone 2
603	Preset Code for the Driver
604	LJM
605	PRS of Driver
POSX 606	POS Subroutine
POS 607	Macro
610	LJM
611	POS.
WDSX 612	WDS Subroutine
WDS 613	Macro
614	LJM
615	WDS.
RDSX 616	RDS Subroutine
RDS 617	Macro
620	LJM
621	RDS.
	POS.
	RDS.
	WDS.
	PRS for Driver
7000	Buffer
7001	

Figure 7-9. MS Driver Core Map

The routine PRS is overlayed with error processing code. This code and the dayfile message is received from MTR in MB - MB+5 after the requesting PP routine issues monitor function DEPM. The code to perform the error processing is loaded at PP location 7510 and is contained in overlay 7DB. The error processing code received from MTR simply issues the dayfile error message unless UERP is set, requests the disk channel, and passes control to the ERX routine. If UERP was not specified, ERX returns to RDS/WDS. If user error processing had been specified via UERP, the error processing code passed from mointor will return control to the caller of RDS/WDS after N retries. Figure 7-10 describes the above operation. The encircled numbers represent the sequence of events.



Description

1. A call to WDS made
2. WDS detects error and calls 7DB.
3. 7DB calls MTR with DEPM function.
4. MTR stores error processing code and return information in message buffer.
5. 7DB moves E.P. code from MB to EXIA, thus overlaying PRS routine in driver.
6. 7DB passes control to that code.
7. E.P code executes and passes control to ERX.
8. If no UERP, retry WDS "n" times. If maximum retries, wait for GO/DROP from operator.

8'. If UERP specified and max. retries, pass control to caller of WDS.

Figure 7-10. Disk I/O Error Control

The following can not happen:
During a read/write on disk, an error occurs, and while trying to issue the error message, an error occurs on the device to which we were writing the error.

Because errors are written to dayfiles and not to devices. When the dayfile buffer is finally written to a device by 1DD, the mass storage driver will not attempt to issue any error messages, but normal corrective action is attempted. If all attempts fail, then the operator is informed via the console dayfile if possible.

The following functions are flowcharted in Figures 7-11 through 7-15.

- PRS - Preset
- STS - Check Status
- POS - Position Disk
- RDS - Read Sector
- WDS - Write Sector

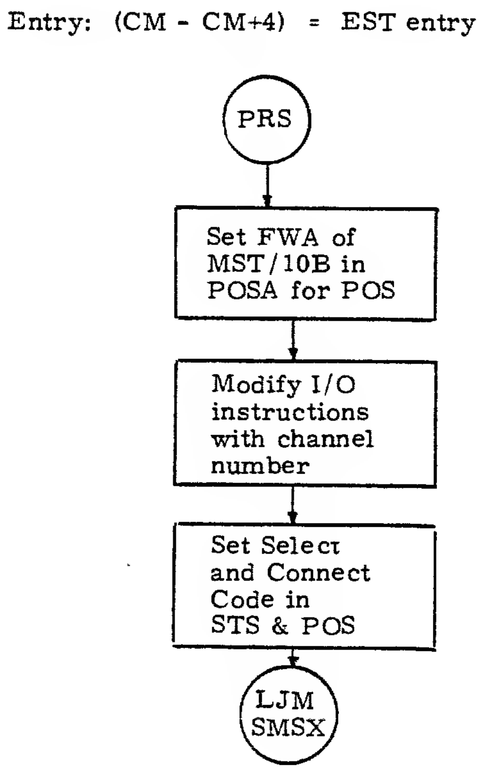


Figure 7-11. PRS - PRESET

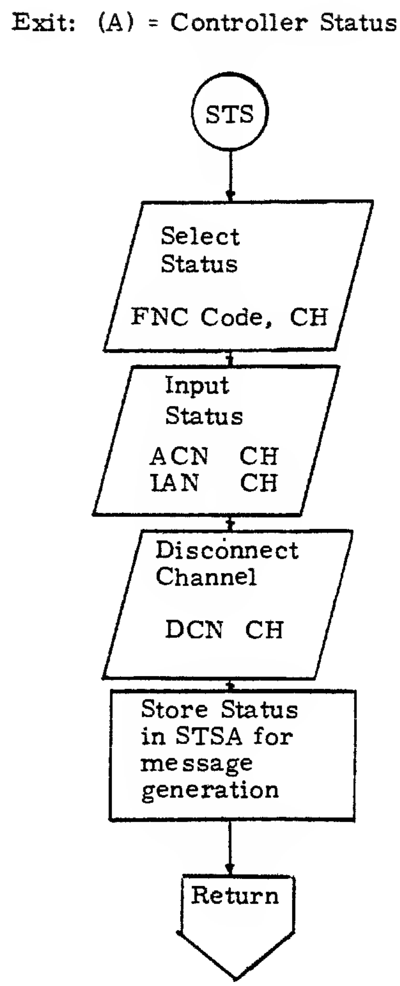


Figure 7-12. STS - Check Status

Entry: (POSA) = FWA/10B of MST entry. Set in PRS.

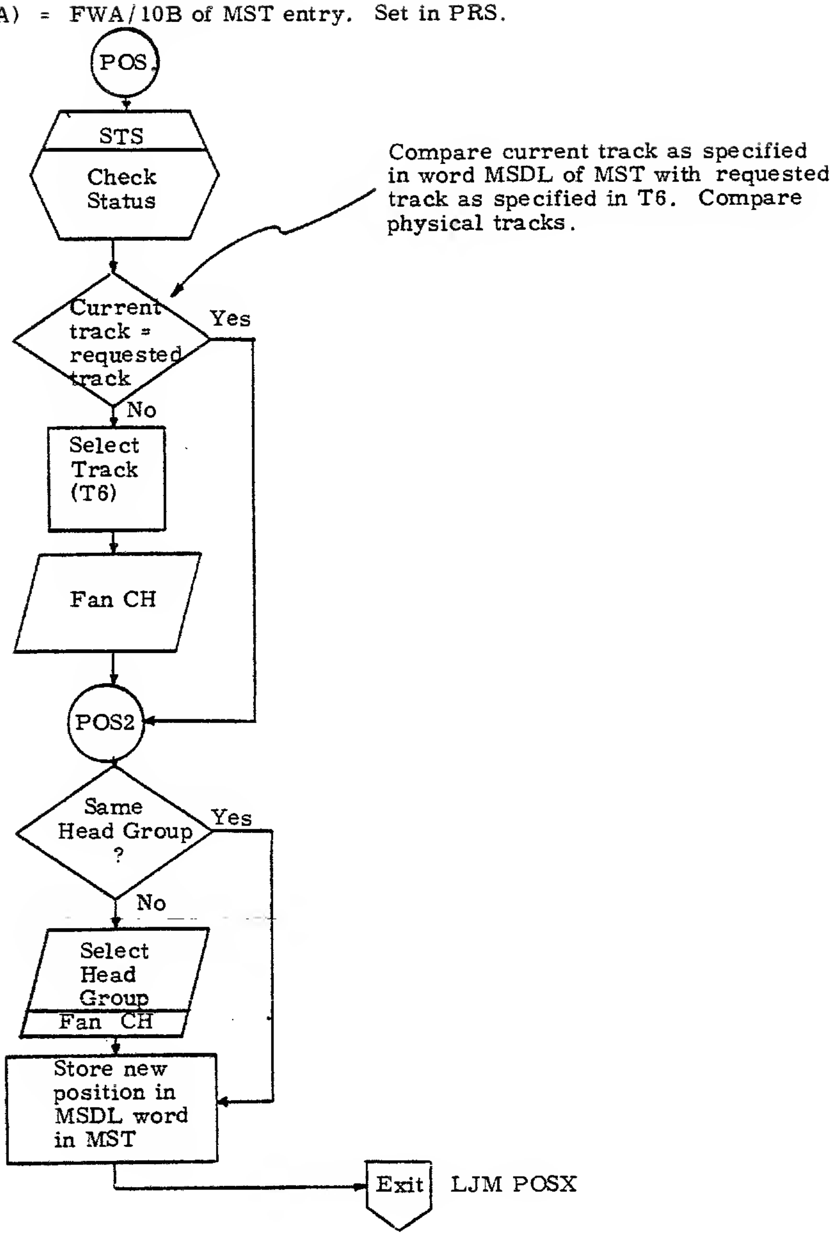


Figure 7-13. POS - Position Disk

Entry: (A) = FWA of PP buffer to contain sector
Exit: (A) < 0 if unrecoverable parity error

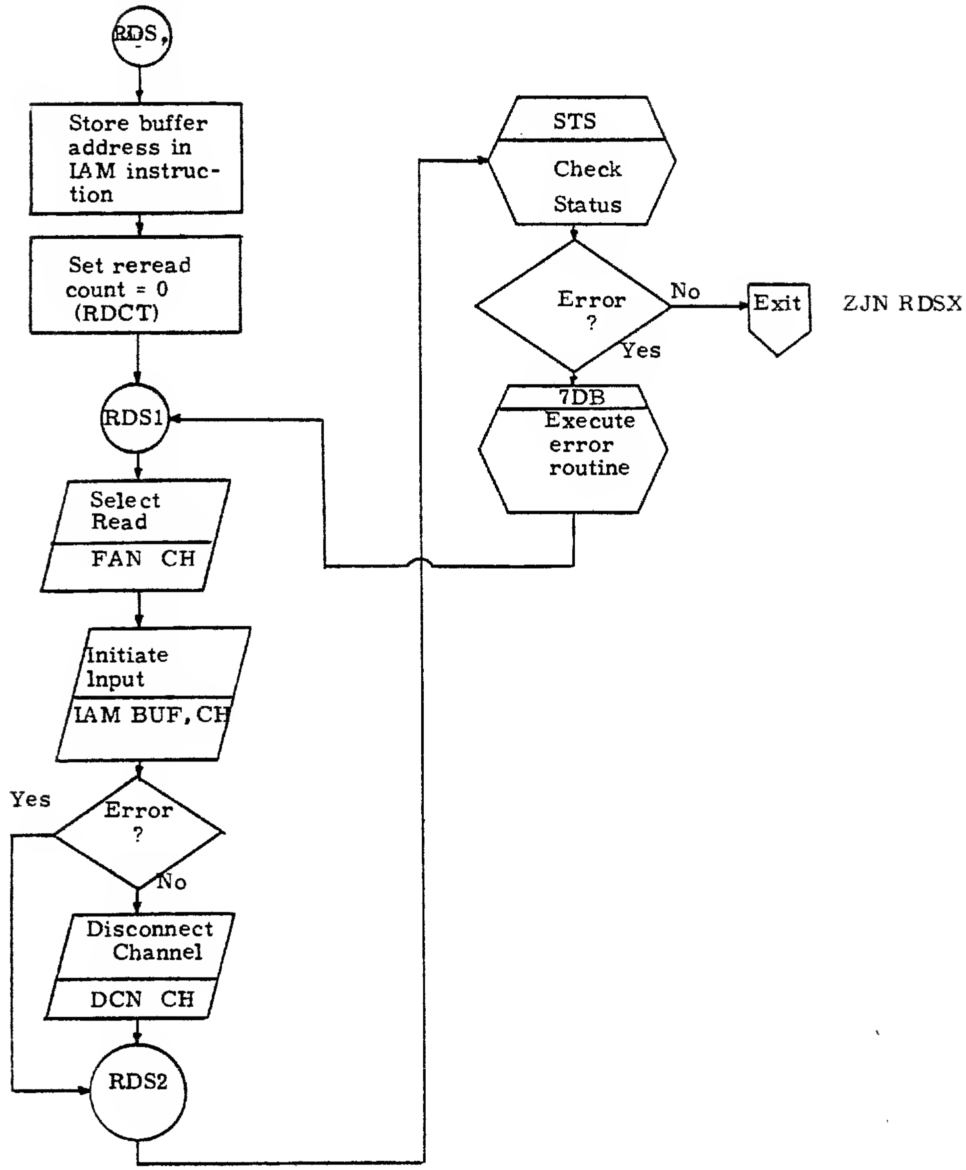
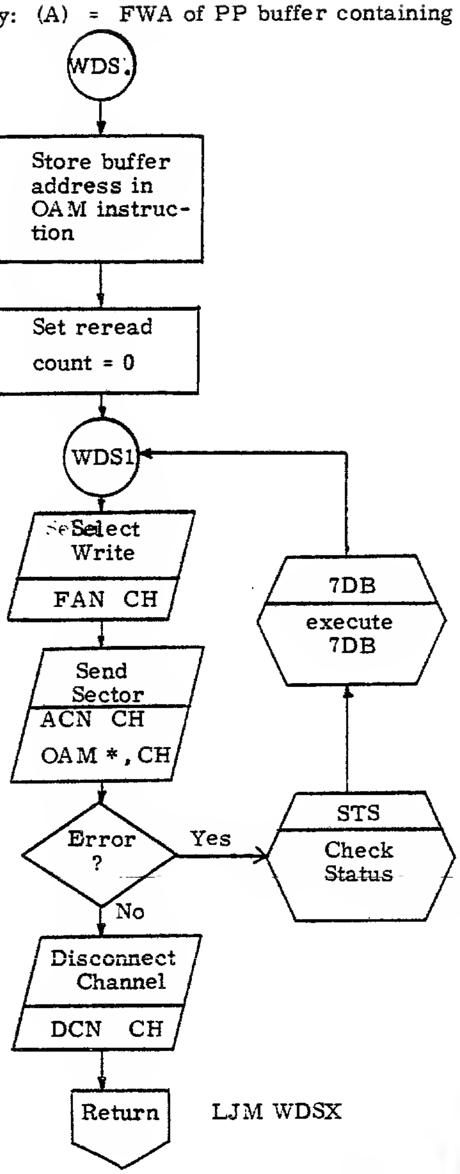


Figure 7-14. RDS - Read Sector

Entry: (A) = FWA of PP buffer containing sector



ERX - Error exit processing

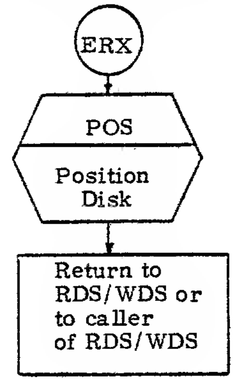


Figure 7-15. WDS - Write Sector

7.6 SYSTEM SECTOR

The system sector is the first sector of a mass storage file and contains system information. PP routines that write mass storage files begin by writing a system sector. Such routines include CIO, 1TA, 1RO, and others. The system sector is generally written via the PP common deck, COMPWSS. Although the calling routine stores various system information in the system sector, COMPWSS stores the control (header) bytes, the FNT/FST, and the date according to the format shown in Figure 7-16. System information varies with different routines. For example, a rollout file's system sector includes dayfile buffer pointers, a copy of the input file's FNT/FST, any operator assigned equipment, and terminal table information for time-sharing jobs.

			3777	0077
FNT entry				
EQ	FT	0000	0000	FA
Packed Date from PDTL in CMR				
system information				

where,

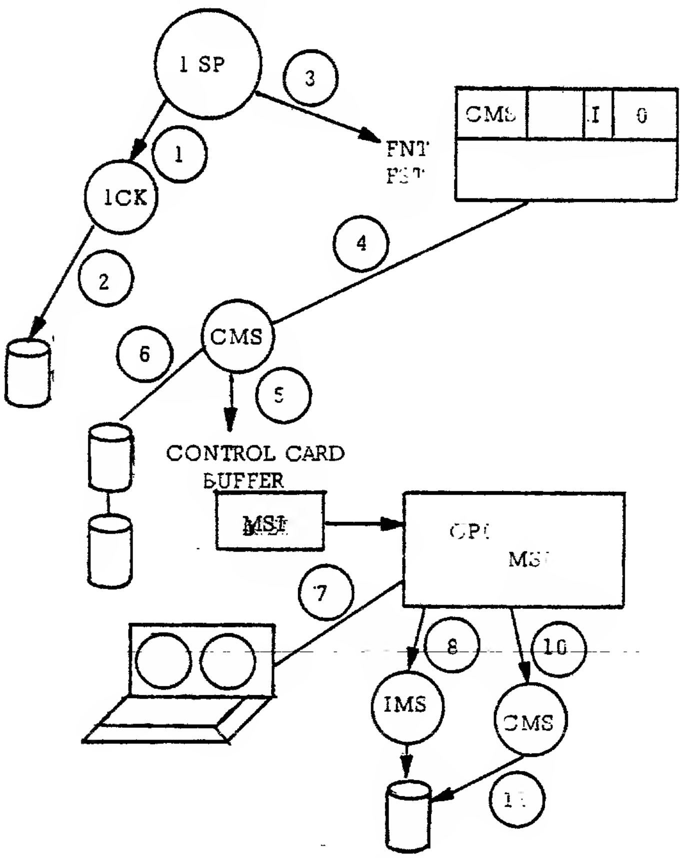
EQ = EST ordinal of this equipment
FT = first track of this file
FA = address of FST entry

Figure 7-16. System Sector Format

Finally, system sectors are useful in untangling tangled-up disk files as time and date will give an idea of when the file was written.

7.7 MASS STORAGE CONTROL AND INITIALIZATION

Mass storage control and initialization is controlled by the system dynamically. ISP will call ICK and CMS periodically according to assembly constants defined in CMR JSCL + 1 word 41 and PFNL + 1 word 111. Figure 7-17 shows this interaction. The other figures show an example of an MST and a LABEL track.



1. 1SP calls 1CK if time period elapsed and checkpoint request is present in MST (every 30 seconds).
2. 1CK will copy the CMR MST/TRT to the label track and drop.
3. 1SP initiates CMS (transient PP) via dummy FNT job entry if:
   a. Initialize request or
   b. Time period elapsed (1 minute)
4. CMS will be started in a PP by 1SP.
5. CMS checks for an Initialize request and if present initiates MSI (via CC call).
6. CMS
   a. Checks labels of active devices and verifies with resident MSTs.
   b. Logically removes inactive devices not physically present (removable only).
   c. Logically receives new devices if label is recognizable:
      - Recover TRTs.
      - Verify PF system.
      - Clean up any DA interlocks.
   d. Drop from PP.
7. MSI interacts with the OPERATOR via K display to set initialize parameters.
8. MSI calls the service PP, IMS.
9. IMS initializes the device(s) and then drops.
10. When IMS is complete, MSI calls CMS and ends (i.e. drops from the CP).
11. CMS verifies the initialized device(s) and its corresponding MST/TRT in CMR and drops.

Figure 7-17. Mass Storage Control and Initialization

TABLE 7-3. DESCRIPTION OF MST

Word	Bits	Description
0	36-59	401205 available sectors
	24-35	624 words in TRT
	11-23	empty
	0-11	2317 TRKs available
1	48-59	N/A
	47	FORMAT PACK not pending
	46	don't release reservation on channel release
	45	Reserved
	36-44	1 unit sec limit = 153
	18-36	sector limit = 153
	0-17	sector limit = 153
2 & 3		reserved for MS drivers
4	48-59	260 is 1st track of Indirect PF chain
	36-47	0 is label track
	24-35	261 is 1st track of permit info
	11-23	catalog track count is 20
	0-11	system table track is 262
5	48-59	7042 is 111 000 100 010 555 555 555 544 bit count 987 654 321 098
	which means	MS device — bit 59 System on device — bit 58 PF on device — bit 57 Direct PF on device — bit 53 TEMP device — bit 49
	24-47	Reserved
	12-23	current direct access files in use = 3 (VALIDUX, RESEXDF, RESEXUF)
	0-11	device in use device not busy bit 0 is on.
6	18-59	Family name is MORRI
	12-17	device number is 40
	11	catalog tracks not continuous with Label Track.
	10	catalog tracks have not overflowed
	8-9	reserved
	0-7	mask is 377.

TABLE 7-3. DESCRIPTION OF MST (Continued)

MST ADDRESS = 4300

```
00401205062400002317
77770153000153000153
00000000000000000000
00000000000000000000
42604000426100204267
70420000000000030005
15172222110000400377
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
```

TABLE 7-4. DESCRIPTION OF LABEL TRACK

Sector number (SE)		Description		
	0	System sector		
Word	-1	Header bytes B1=37777, B2=77		
	0	FNT see p. 3-19 of instant		
	1	Eqss = eq = 0, 1st track = 0		
	2	Date = July 2, 1974, time 12.14.56, updated at every level 0 Dead Start		
	3,4,5,6,7	Empty all part of standard system sector		
	10,11	Empty		
	12,13	Used for 2.0 PF compatability		
	14	Catalog descriptor entry		
			Bits 0-11	Empty
			12-23	Mask
			24-35	Number of catalog tracks
			36-60	Empty
	15	Track descriptor		
			Bits 0-11	Empty
			12-33	1st track permit
			24-35	Empty
			36-47	1st track of indirect file chain
			48-60	Empty
	16		Bits 0-23	Empty
			24-35	Sector number for continuation of label track (all tracks) on next spindle for multi-unit devices.
			36-47	Sector limit
			48-60	Device type

TABLE 7-4. DESCRIPTION OF LABEL TRACK (Continued)

Sector number (SE)	Description
17	K 2.0 compatability Bits 0-35 Empty 36-47 Word count of mass storage table = 20 words 48-60 Catalog ordinal biased by 40
20	EST entry
21-30	1st 10 words of MST
31	Password for removable device
32-77	Unused
1 + however many needed	copy of TRT
Foil	**Description**
1	Listing of label tracks documentation from COMSDSL.
2	DUMPTK of LABEL from DI-1 eq 0, SYSTEM and PF device.
3	Continuation of 2.

CALLSYS - CALL SUB-SYSTEM COMMON DECKS. 73/08/29. 11.57.30.
COMSDSL - DEAD START LOAD PARAMETERS.

TABLE 7-5. DEFINITIONS FOR DEVICE LABEL SECTOR.*

Byte Number	Word Number	Definitions			
7026		.1	SET	BFMS+2+4*5	
7076	14	CESS	EQU	.1+10*5	CATALOG DESCRIPTOR ENTRY
7100		CCSS	EQU	.1+10*5+2	CATALOG COUNT
7101		CDSS	EQU	.1+10*5+3	DEVICE MASK
7104	15	DASS	EQU	.1+11*5+1	FIRST TRACK OF INDIRECT FILES
7105		ALSS	EQU	.1+11*5+2	LABEL TRACK (LINKED TO CATALOG TRACKS).
7106		PRSS	EQU	.1+11*5+3	FIRST TRACK OF PERMITS.
7110	16	ETSS	EQU	.1+12*5	EQUIPMENT TYPE
7111		SLSS	EQU	.1+12*5+1	SECTOR LIMIT
7112		SNSS	EQU	.1+12*5+2	SECTOR NUMBER FOR MULTI UNIT DEVICES
	*	THE FOLLOWING 2 WORDS ARE USED FOR PURPOSES OF FUTURE			
	*	COMPATIBILITY.			
7115	17	COSS	EQU	.1+13*5	CATALOG ORDINAL (BIASED BY 40)
7116		WCSS	EQU	.1+13*5+1	WORD COUNT OF MASS STORAGE TABLE
7122	20	ESSS	EQU	.1+14*5	EST ENTRY
7127	21	MTSS	EQU	.1+15*5	FIRST WORD OF MASS STORAGE TABLE
7177	31	PWSS	EQU	.1+25*5	PASSWORD FOR REMOVABLE DEVICE
7204		LMSS	EQU	.1+26*5	LIMIT OF CHECK OF LABEL DATA

These are in byte counts:

.1 equ 4 so

CESS is 4 + 10 = word 14 *5 = byte 0
CCSS is 4 + 10 = word 14 *5 + 2 = byte 2
CDSS is 4 + 10 = word 14 *5 + 3 = byte 3
DASS is 4 + 11 = word 15 *5 + 1 = byte 1

etc.

*This information is from common deck COMSDSL.

DUMPTK(TK=-)   DUMPTK - VER. 1   74/07/02. 11.14.07.   PAGE 1

WORD	TK=0 SF=0 B1=3777 B2=77		TK=0 SE=1 B1=2	B2=100	TK=0 SE=2 B1=3	B2=100
0	14010705140000010500	LABEL AE	42374002400340044017	745H5C5050	44014402440344040017	9A9B9C9D 0
1	00004007000000000000	5	40054006400740100017	5E5F5G5H 0	44054406440744100017	9E9F9G9H 0
2	00000000040703131456	OGHKL,	40114012401340140017	515J5K5L 0	44114412441344140017	919J9K9L 0
3	00000000000000000000		40154016401740200017	5M5N5O5P 0	44154416441744200017	9M9N9O9P 0
4	00000000000000000000		40214022402340240017	5Q5R5S5T 0	44214422442344240017	9Q9R9S9T 0
5	00000000000000000000		40254026402740300017	5U5V5W5X 0	44254426442744300017	9U9V9W9X 0
6	00000000000000000000		40314032403340340017	5Y5Z5051 0	44314432443144340017	9Y9Z9091 0
7	00000000000000000000		40354036403740400017	52535455 0	44363777444037770017	93419541 0
10	00000000000000000000		40414042404340440017	56575859 0	44414442444344440017	96979899 0
11	00000000000000000000		40454046404740500017	5+5-5*5/ 0	44454446444744500017	9+9-9*9/ 0
12	00000000000000000000		40514052405340540017	5(5)5$5= 0	44514452445344540017	9(9)9$9= 0
13	00000000000000000000		40554056405740600017	5 5,5.5≡ 0	44554456445744600017	9 9,9.9≡ 0
14	00000000120173770000	PC)	40614062406340640017	5[5]5:5≠ 0	44614462446344640017	9[9]9:9≠ 0
15	00004260004002610000	7= 7,	40654066406740700017	5→5∨5∧5↑ 0	44654466446744700017	9→9∨9∧9↑ 0
16	04110150000000000000	U1A$	40714072407340740017	5↓5<5>5≤ 0	44714472447344740017	9↓9<9>9≤ 0
17	00000020000000000000	,	40754076407741000017	5≥5¬5;6 0	44754476447745000017	9≥9¬9;+ 0
20	70000002000104110430	+ = A11X	41014102410341040017	6A6B6C6D 0	45014502450345040017	+A+B+C+D 0
21	00343300062400002100	10 FT 0	41054106410741100017	6E6F6G6H 0	45054506450745100017	+E+F+G+H 0
22	77770151000153000153	1144 A$ A$	41114112411341140017	616J6K6L 0	45114512451345140017	+I+J+K+L 0
23	00000000000000000000		41154116411741200017	6M6N6O6P 0	45154516451745200017	+M+N+O+P 0
24	00000000000000000000		41214122412341240017	6Q6R6S6T 0	45214522452345240017	+Q+R+S+T 0
25	42604700420100204202	7≡5 71 07J	41254126412741300017	6U6V6W6X 0	45254526452745300017	+U+V+W+X 0
26	70420000000000010005	+7 C E	41314132413341340017	6Y6Z6061 0	45314532453345340017	+Y+Z+0+1 0
27	15172222110000400377	HORMI SC,	41354136413741400017	62636465 0	45354536453745400017	+2+3+4+5 0
30	00000000000000000000		41414142414341440017	66676869 0	45414542454345440017	+6+7+8+9 0
31	00000000000000000000		41454146414741500017	6+6-6*6/ 0	45454546454745500017	+++-+*+/ 0
32	00000000000000000000		41514152415341540017	6(6)6$6= 0	45514552455300570017	+(+)+$. 0
33	00000000000000000000		41554156415741600017	6 6,6.6≡ 0	45700000000045600411	++ +≡D1
34	00000000000000000000		41614162416341640017	6[6]6:6≠ 0	45614562456345670017	+[+]+:+∧ 0
35	00000000000000000000		41654166416741700017	6→6∨6∧6↑ 0	45654577000045740015	+→+) +S M
36	00000000000000000000		41714172417341740017	6↓6<6>6≤ 0	45734572455400440017	+>+<+= 9 0
37	00000000000000000000		41754176417742000017	6≥6¬6;7 0	45754576460246000017	+≥+¬-A- 0
40	00000000000000000000		42014202420342040017	7A7B7C7D 0	46014613460346040017	-A-K-C-D 0
41	00000000000000000000		42054206420742100017	7E7F7G7H 0	46054606461246100017	-F-F-J-H 0
42	00000000000000000000		42114212421342140017	717J7K7L 0	46114614460743440017	-I-L-GA9 0
43	00000000000000000000		42154216421742200017	7M7N7O7P 0	46154616461746200017	-M-N-O-P 0
44	00000000000000000000		42214222422342240017	7Q7R7S7T 0	46214622462346240017	-Q-R-S-T 0
45	00000000000000000000		42250065422742300017	7U →7W7X 0	46314626462746300017	-Y-V-W-X 0
46	00000000000000000000		42314232422330035017	7Y7Z7n 2 0	46324625463346340017	-Z-U-0-1 0
47	00000000000000000000		00010001000142400017	A A A75 0	46354636463746400017	-2-3-4-5 0
50	00000000000000000000		42414242424342440017	76777879 0	46414642466346440017	-6-7-8-9 0
51	00000000000000000000		42454246424742500017	7+7-7*7/ 0	46454646464746500017	-+---*-/ 0
52	00000000000000000000		42514252425342540017	7(7)7$7= 0	46514652465346540017	-(-)-$-= 0
53	00000000000000000000		42554256425700000017	7 7,7. 0	46554656465746600017	- -,-.-≡ 0
54	00000000000000000000		45640001427142706417	+≠ A7+7+≠0	46614662466346640017	-[-]-:-≠ 0
55	00000000000000000000		00050044004000040717	F 9 5 5≤0	46654666466746700017	-→-∨-∧-↑ 0
56	00000000000000000000		42720023427342740017	7< 57>7≤ 0	46714672467346740017	-↓-<->-≤ 0
57	00000000000000000000		42754276427743000017	7≥7¬7;8 0	46754676467747000017	-≥-¬-;* 0
60	00000000000000000000		43014303000243040017	8ABC HHD 0	47014702470347040017	*A*B*C*D 0
61	00000000000000000000		43054306430743100017	8E8F8G8H 0	47054706470747100017	*E*F*G*H 0
62	00000000000000000000		43114312431343140017	8I8J8K8L 0	47114712471347140017	*I*J*K*L 0
63	00000000000000000000		43154316431743200017	8M8N8O8P 0	47154716471747200017	*M*N*O*P 0
64	00000000000000000000		43214322432343240017	8Q8R8S8T 0	47214722472347240017	*Q*R*S*T 0
65	00000000000000000000		43254326432743300017	8U8V8W8X 0	47254726472747300017	*U*V*W*X 0
66	00000000000000000000		43314332433343340017	8Y8Z8081 0	47314732473347340017	*Y*Z*0*1 0
67	00000000000000000000		43354336433743400017	82838485 0	47354736473747400017	*2*3*4*5 0

```
        DUMPTK(TK=0)                                                   DUMPTK - VER. 1      74/07/02. 11.14.07.   PAGE   2

 70     00000000000000000000                 43414342434343450017  86878A8* 0    47414742474347440017  *6*7*8*9 0
 71     00000000000000000000                 43714346434743500017  *+8-8*8/ 0    47454746474747500017  ***-***/ 0
 72     00000000000000000000                 43514352435343540017  8(8)8$8= 0    47514752475347540017  *(*)*$** 0
 73     00000000000000000000                 43554356435743600017  8 8,8.8= 0    47554756475747600017  * *,*.*= 0
 74     00000000000000000000                 43614362436343640017  8[8]8:8* 0    47614762476347640017  *[*]*:*# 0
 75     00000000000000000000                 43654366436743700017  8+8v8^8+ 0    A7654766476747700017  *+*v*^*+ 0
 76     00000000000000000000                 43714372437343740017  8+8<8>8$ 0    47714772477347740017  *+*<*>*$ 0
 77     00000000000000000000                 43754376437744000017  8=8-819  0    47754776477750000017  *=*-*1/  0

WORD    TK=0     SF=3     B1=4     B2=100    TK=0     SE=4     B1=5     B2=100    TK=0     SE=5     B1=6     B2=100
 0      50015002500350040017  /A/B/C/D 0     00000000000000000000                 00000000000000000000
 1      50055006500750100017  /E/F/G/H 0     00000000000000000000                 00000000000000000000
 2      50115012501350140017  /I/J/K/L 0     00000000000000000000                 00000000000000000000
 3      50155014006300000016  /M/N I   N     00000000000000000000                 00000000000000000000
 36     00000000000000000000                 00000000000000000000                 00000000000037770001        41 A
 37     00000000000000000000                 00000000000000000000                 00000000000037770001        A1 A
 42     00000000000000000000                 00000000000000000000                 00000000000037770001        41 A
 47     00000000377700000002        41   0   00000000000000000000                 00000000000000000000  EOF

WORD    TK=0     SF=6     B1=7     B2=100    TK=0     SE=7     B1=10    B2=24     TK=0     SE=10    B1=0     B2=0
 0      00000000000000000000                 00000000000000000000                 01001512303210711207  * HJXZH+JA
 1      00000000000000000000                 00000000000000000000                 04723464303410031601  D<I*XIHCNA
 2      00000000000000000000                 00000000000000000000                 60103011220007773465  =HXIR G$I*
 3      00000000000000000000                 00000000000000000000                 14003407376407553065  L IG4*G X*
 4      00000000000000000000                 00000000000000000000                 35075400711220007000  ?G= +JP +
 5      00000000000000000000                 00000000000000000000                 02000612036501001645  A FJC*A N+
 6      00000000000000000000                 00000000000000000000                 30061074120134013007  XFHSJAIAXG
 7      00000000000000000000                 00000000000000000000                 16015400243553010601  NA= T2$AFA
 10     00000000000000000000                 00000000000000000000                 05143413300534113006  FLIKXFIIXF
 11     00000000000000000000                 00000000000000000000                 34121455020003643014  IJL A C*XL
 12     00000000000000000000                 00000000000000000000                 54002435200024350200  = T2P T2B
 13     00000000000000000000                 00000000000000000000                 06125000243534071006  F.I/ T2IGHF
 14     00000000000000000000                 00000000000000000000                 06071071340614003407  FGH+IFL IB
 15     00000000000000000000                 00000000000000000000                 02000606010015460100  A FFA M-A
 16     00000000000000000000                 00000000000000000000                 21763467107134663071  Q-IAH+IVX+
 17     00000000000000000000                 00000000000000000000                 54002436306607660505  = T3XVOVEE
 20     00000000000000000000                 00000000000000000000                 30671277540024363064  XA.I*= T3X*
 21     00000000000000000000                 00000000000000000000                 10143116561712437020  HLY*[+T4B
 22     00000000000000000000                 00000000000000000000                 15473071356510633564  M*X+2*H12*
 23     00000000000000000000                 00000000000000000000                 37660355010016533030  4VC A N$XX
 24     00000000000000000000                 00000000000000000000                 10070673303410031604  HGF>XIHCND
 25     00000000000000000000                 00000000000000000000                 60103014040734123005  =HXLOGIJXF
 26     00000000000000000000                 00000000000000000000                 34111443020003643005  IIL AB C*XF
 27     00000000000000000000                 00000000000000000000                 34111400341234131455  IIL IJIKL
 30     00000000000000000000                 00000000062400003120       FT   YP    02000364301434063436  A C*XLIFI3
 31     00000000000000000000                 77770153000153000153  IIA$ A$ A$       30341003160460103036  XIHCND=HX3
 32     00000000000000000000                 00000000000000000000                 34143034100316046210  ILXIHCNDIH
 33     00000000000000000000                 00000000000000000000                 30776370111717016170  X$I+IOOA(+
 34     00000000000000000000                 00000000000000000000                 70021400540070133457  +BL = +KI.
 35     00000000000000000000                 44100000000000000001  9H          A    02002327140034643465  A SWL I*I*
 36     00000000000000000000                 00000000000000000000                 14500200162030603465  (/A NPX=I*
 37     00000000000000000000                 00000000000000001300            K      30613260020016203062  XI7=B NPXI
 40     00000000000000000000                 00000000000000000000                 34653063326202001620  I*XIZIA NP
 41     00000000000000000000                 00000000000000000000                 14036020302034653025  ICEPXPI*XU
 42     00000000000000000000                 00000000000000000000                 17013124100102001620  OAYYHAR NP
 43     00000000000000000000                 00000000000000000000                 37240717040431251701  $TGOOOYUOA
 44     00000000000000000000                 00000000000000000000                 10013120601030103464  HAYP=HXHI*
 45     00000000000000000000                 00000000000000000000                 30113465301202001620  XII*XJA NP
```

97404700C

8.0 INTRODUCTION

Combined Input/Output (CIO) processes input/output requests for CPU programs. Data transfer between CIO and the CPU program is handled via a buffer within the CPU program's field length. This buffer is known as a circular buffer because CIO treats the last word and the first word as contiguous. The circular buffer is controlled via a File Environment Table (FET) which is also within the job's field length. The FET not only describes the buffer, but also holds the request code being issued to CIO. Figure 8-1 shows the relationship between CIO, the FET, and the circular buffer. For a write operation, at least one PRU of data should be in the buffer. For a read operation, the buffer must have room to receive one PRU of data. Less than one PRU of data is transferred only if an End-Of-Record (EOR) is read or written.

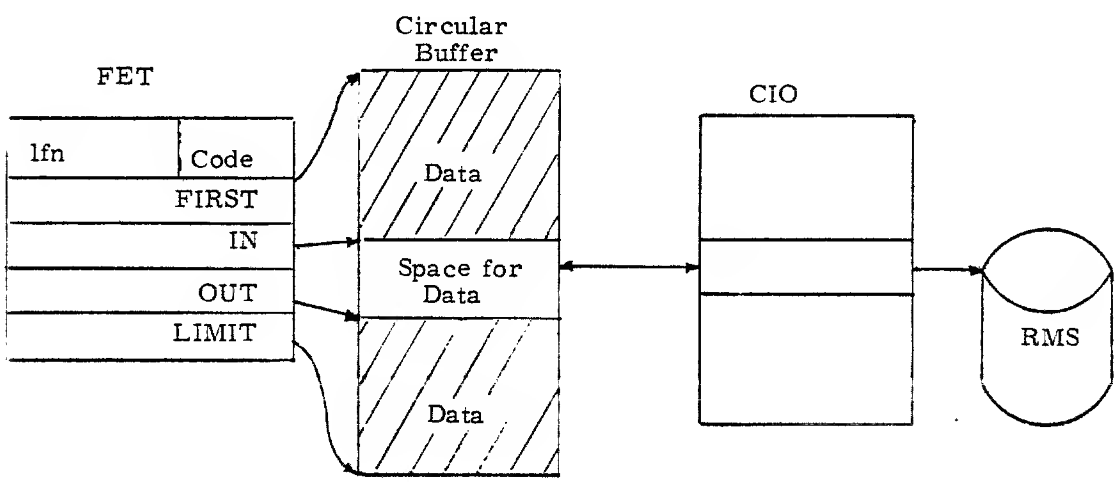


Figure 8-1. User/CIO Interface

The FET formats for mass storage and magnetic tape files are shown in detail in Section 7 (Figures 7-4 and 7-5) of the KRONOS 2.1 Reference Manual.

Equipment which may be accessed by CIO include:

- Mass Storage MS
- Magnetic Tape MT

- Card Reader CR
- Card Punch CP
- Line Printer LP
- Line Printer (512) LQ

Routines used by CIO include:

- 0BF - begin file
- 0DF - drop file
- 2LP - write line printer
- 2PC - write card punch
- 2RC - read card reader

The only routine called by CIO is

- DMP - dump memory

CIO consists of the following overlays:

- CIO - main routine and termination
- 2CA - identify special request
- 2CB - read mass storage
- 2CC - special mass storage reads
- 2CD - write mass storage
- 2CE - special mass storage writes
- 2CF - position mass storage
- 2CG - terminal input/output
- 2CH - magnetic tape operations
- 2CI - error processing
- 2CJ - device error processor

The call to CIO is formatted as follows:

59 48	47 42	41	40	39 36	35 18	17 0
C I	O	0	R	0	skip count	FET address

R = 1 if auto-recall is desired

CIO Memory Map

Figure 8-2 describes PP memory as allocated by CIO. The symbol MSDO is the origination address (ORG) for the mass storage drivers, 2CB and 2CD. The symbol DRFW is the load address for overlay 2CG and for drivers 2LP, 2PC, and 2RC. The symbol OVL is the load address for overlays 2CA and 2CH and for zero level overlays, 0BF and 0DF. The symbol ERPO is the load address of the error processing overlays, 2CI and 2CJ. ERPO follows the last word of the longest overlay, namely, 2CF. Boxes to the right of CIO represent the various overlays called by CIO and their relative lengths. Not shown are any of the overlays and drivers loaded at DRFW. These include 2CG, 2LP, 2PC, and 2RC as stated above. 0BF and 0DF are also not shown. CIO routines are shown in greater detail in Figure 8-3. The logic flow-through CIO is shown in Figure 8-4.

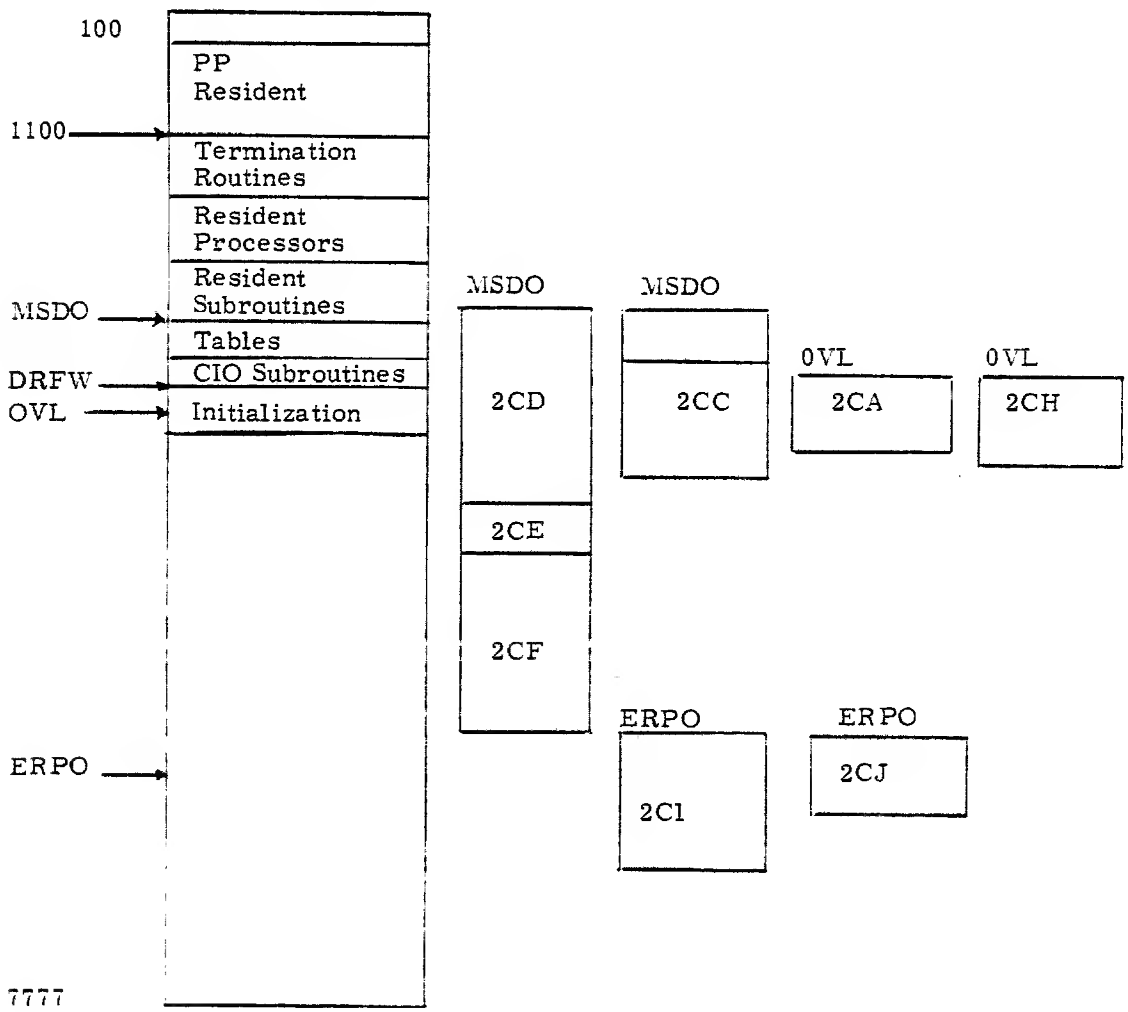


Figure 8-2. CIO Memory Map

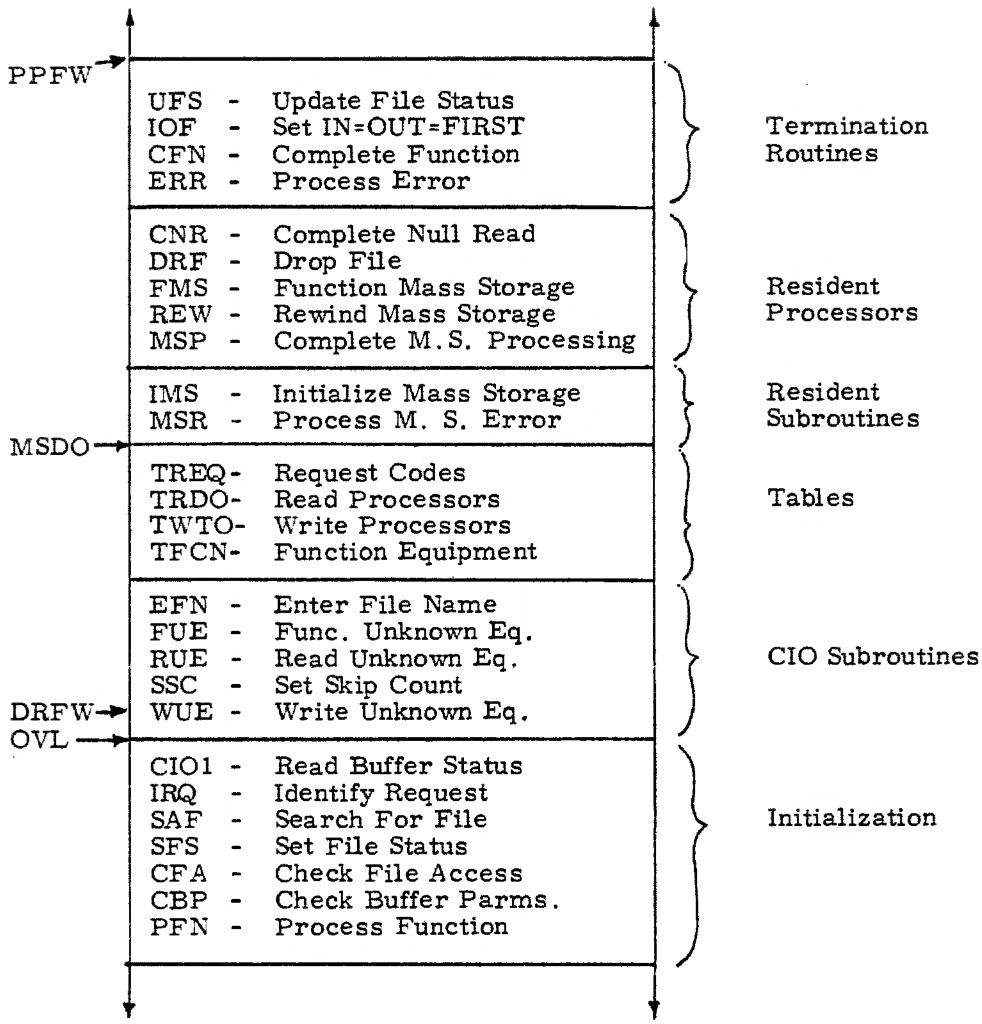


Figure 8-3. CIO - Main Overlay

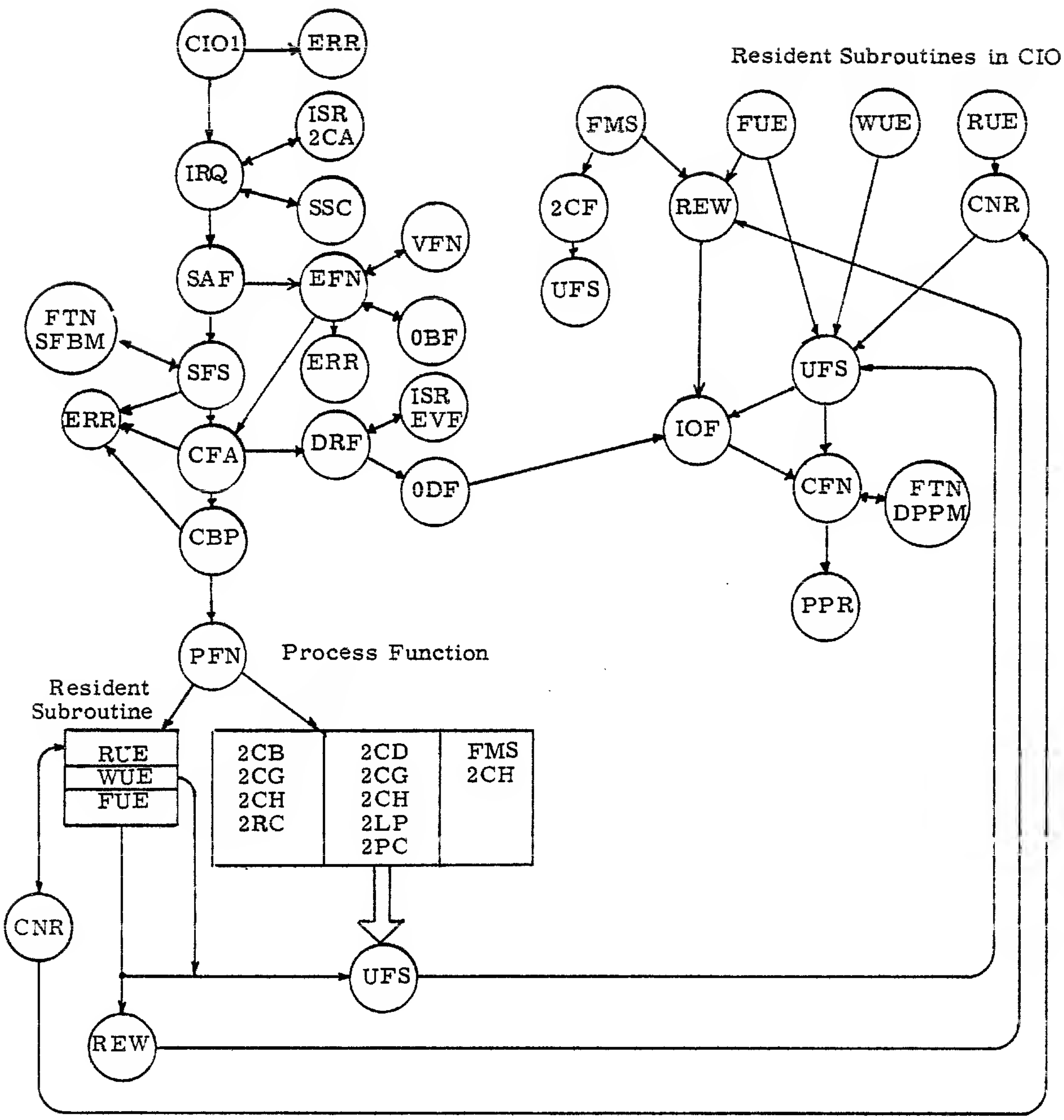


Figure 8-4. CIO Logic Flow

8.1 CIO INITIALIZATION ROUTINES

Figures 8-5 through 8-10 are flowcharts for the following CIO initialization routines:

CIO1/IRQ

SAF

EFN/SFS

CFA

CBP

PFN

The PFN routine searches one of three tables (TRDO, TWTO, or TFCN) to get the name of the overlay to be executed. The three tables are formatted as shown in Table 8-1 through 8-3.

TABLE 8-1. TRDO - TABLE OF READ PROCESSORS

Equipment	Entry Point	Overlay Name
MS	RMS	2CB
TT	TIO	2CG
MT	PMT	2CH
NT	PMT	2CH
CR		2RC
0	RUE	(Read unknown equipment)

TABLE 8-2. TWTO - TABLE OF WRITE PROCESSORS

Equipment	Entry Point	Overlay Name
MS	WMS	2CD
TT	TIO	2CG
MT	PMT	2CH
NT	PMT	2CH
LP		2LP
LQ		2LQ
CP		2PC
0	WUE	(write unknown equipment)

TABLE 8-3. TFCN - TABLE OF FUNCTION PROCESSORS

Equipment	Entry Point	Overlay Name
MS	FMS	(resident)
MT	PMT	2CH
NT	PMT	2CH
0	FUE	(function unknown equipment)

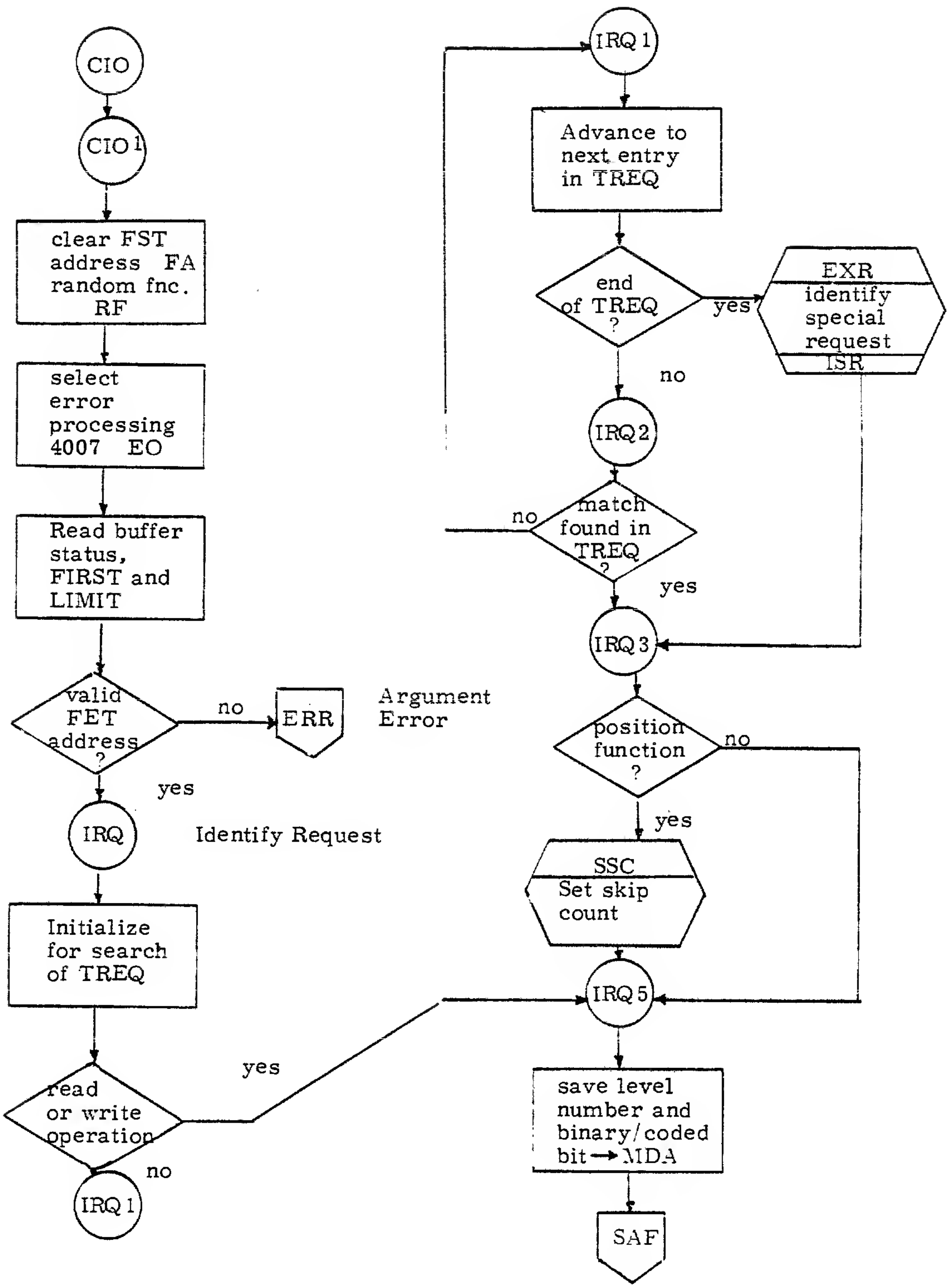


Figure 8-5. CIO Initialization

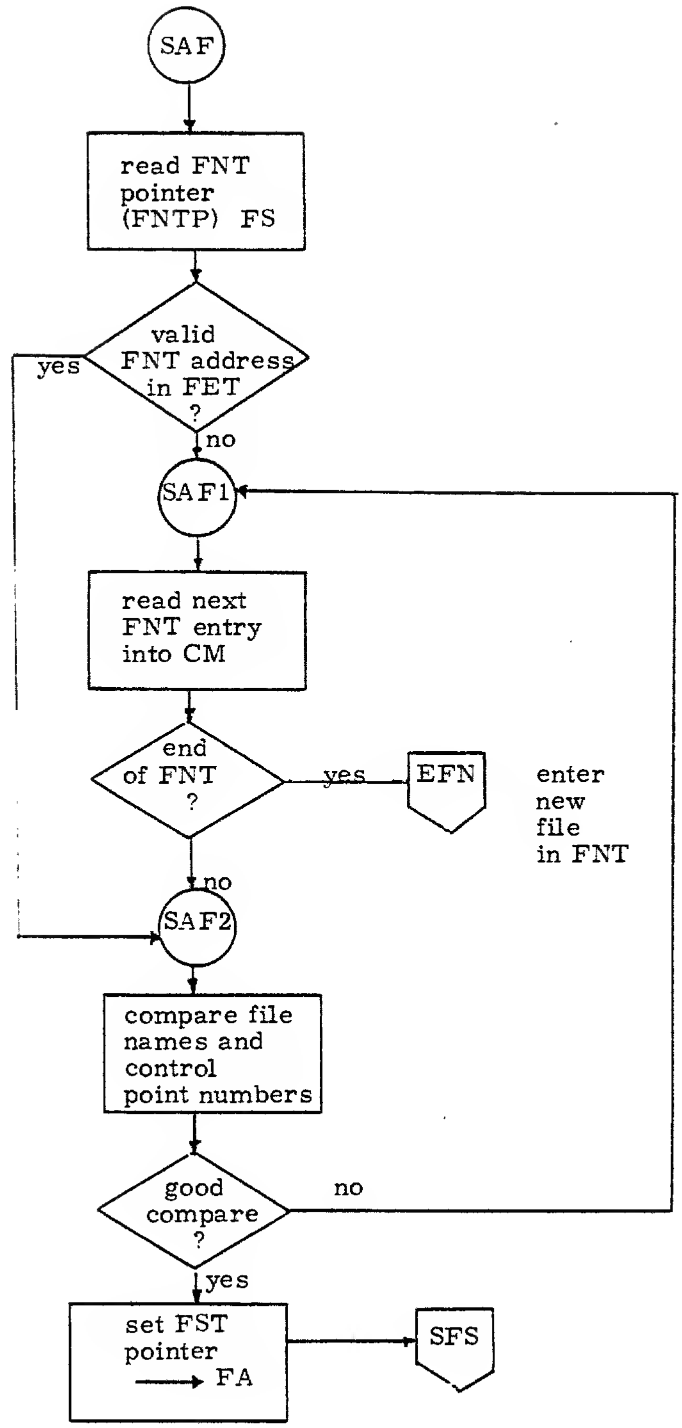


Figure 8-6. SAF - Search for Assigned File

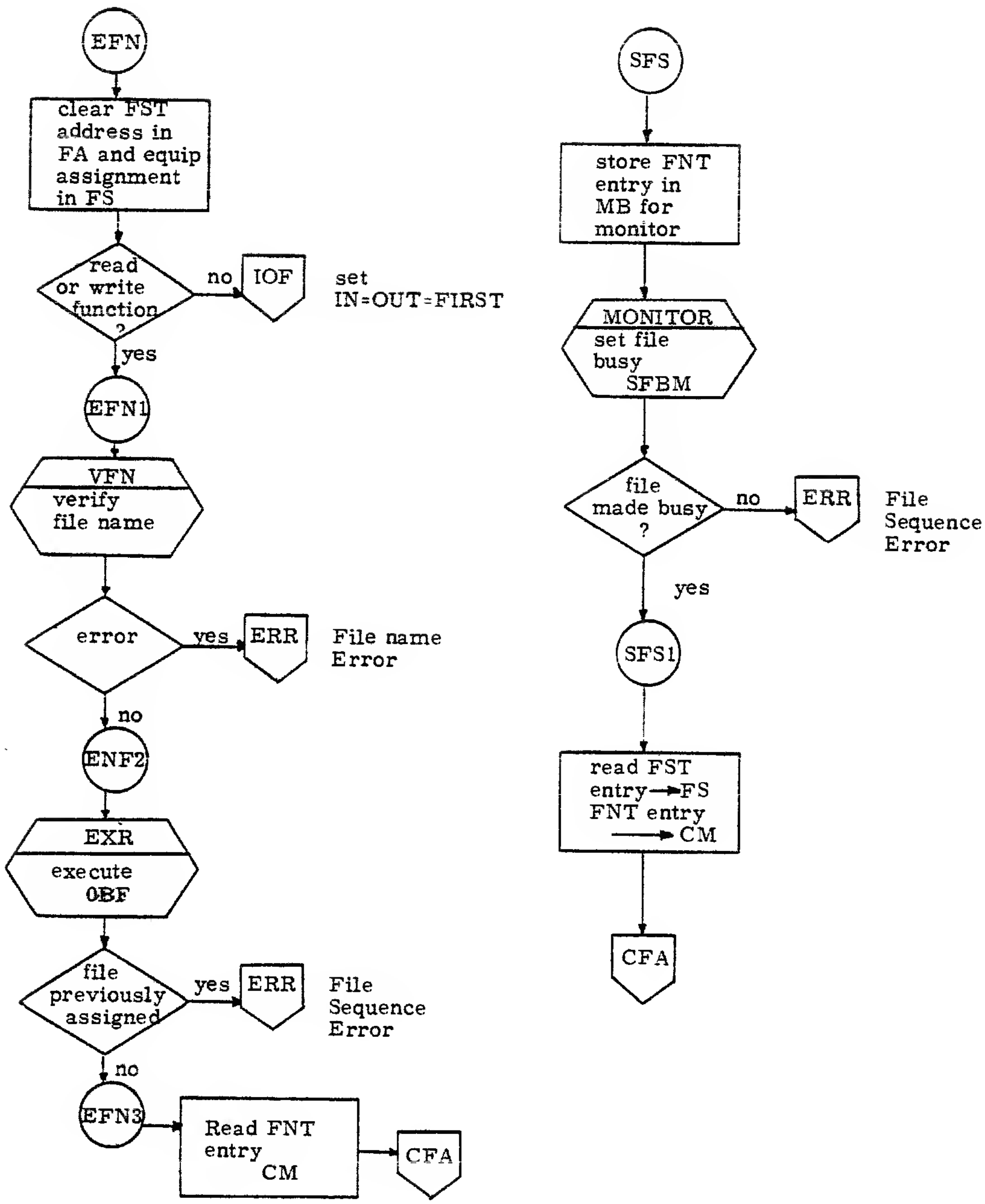


Figure 8-7. EFN - Enter File Name and SFS - Set File Status

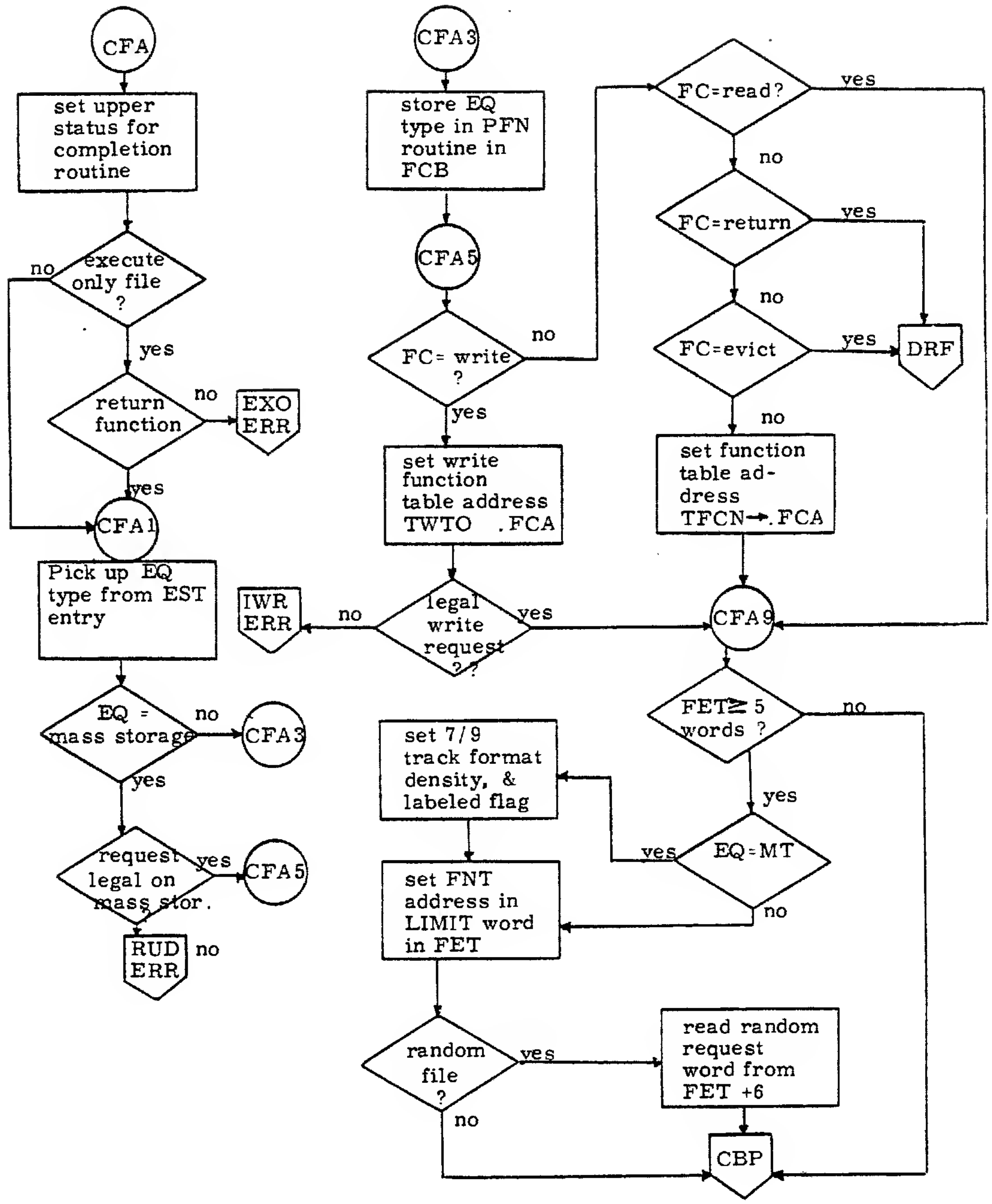


Figure 8-8. CFA - Check File Access

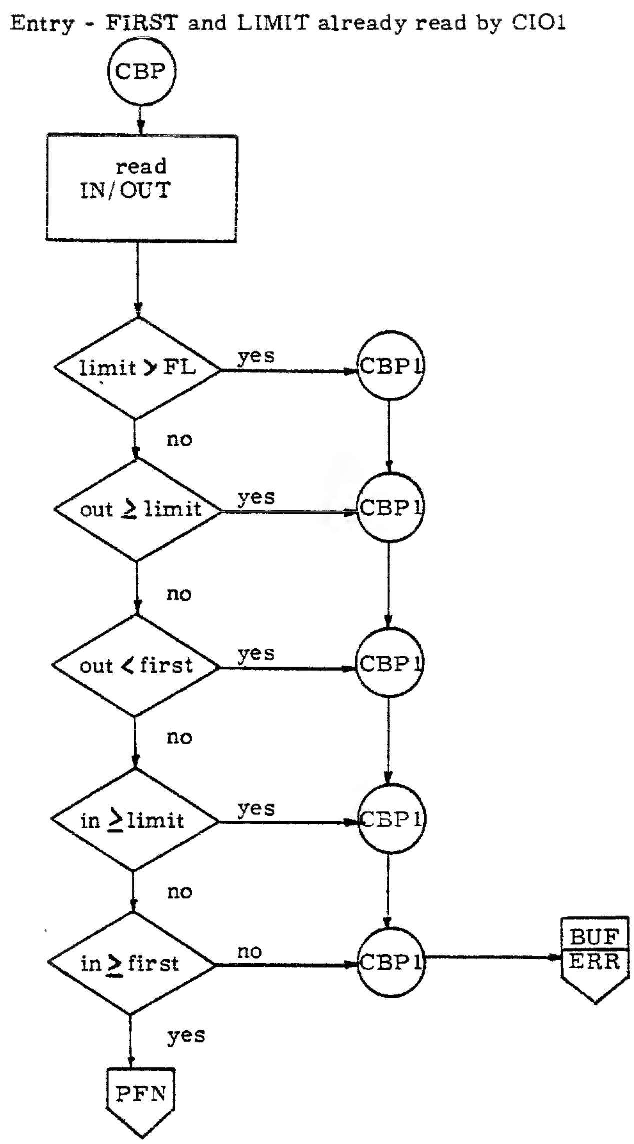
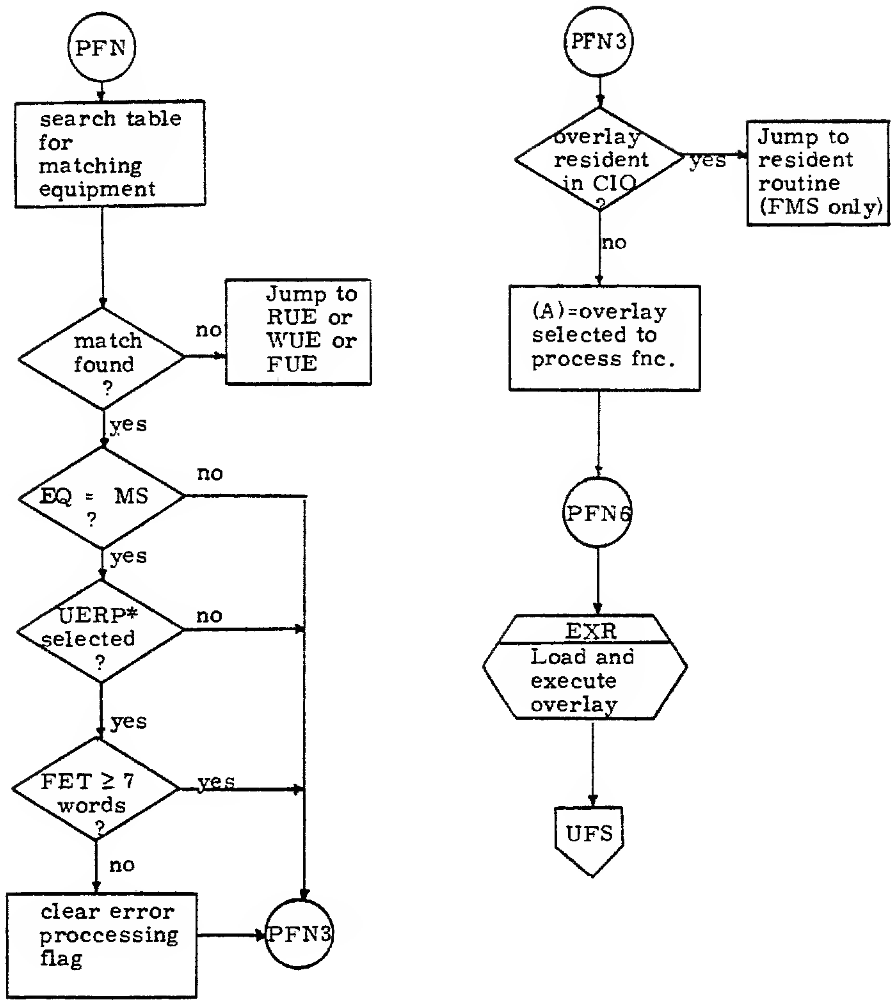


Figure 8-9. CBP - Check Buffer Parameters

Entry - CFA has set:
- FCA = TRDO, TWTO or TFCN
- FCB = Equipment type from EST

Exit - Jump to routine selected from table



* UERP = User Error Processing

Figure 8-10. PFN - Process Function

8.2 CIO ERROR MESSAGES AND ROUTINES

Error messages from CIO are numbered and identified by a unique three character name. Subroutines issuing an error message do so with the following code:

```
LDN    |ERR| XXX
LJM     ERR
```

where XXX is the unique name.

All error messages are in overlay 2CI Table 8-4.

TABLE 8-4. OVERLAY 2CI

Name	Message
ARG	FET ADDRESS OUT OF RANGE
BLE	BUFFER CONTROL WORD ERROR ON
BUF	BUFFER ARGUMENT ERROR ON
DRE	DEVICE ERROR ON FILE
EXO	I/O ON EXECUTE ONLY FILE
FLN	ILLEGAL FILE NAME
FSQ	I/O SEQUENCE ERROR ON FILE
IFE	ILLEGAL EXTENSION OF
IFM	ILLEGAL MODIFICATION OF
IRQ	ILLEGAL I/O REQUEST ON FILE
IWR	WRITE ON READ ONLY FILE
RAD	RANDOM ADDRESS NOT ON FILE
RUD	REQUEST UNDEFINED ON DEVICE
RWT	INDEX ADDRESS OUT OF RANGE FOR
TKL	TRACK LIMIT, FILE
TNA	M.T. NOT AVAILABLE ON FILE

The logical file name and FET address follow the above messages. The error processing subroutine ERR is flowcharted in Figure 8-11 and the overlay 2CI called by ERR is flowcharted in Figure 8-12.

Entry - (A) = Error Number

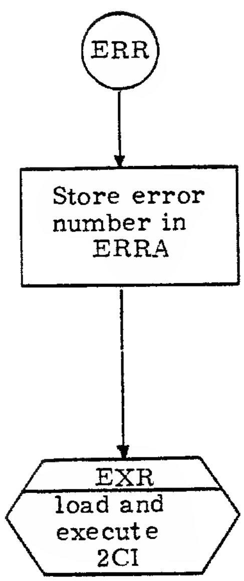


Figure 8-11. ERR - Process Error

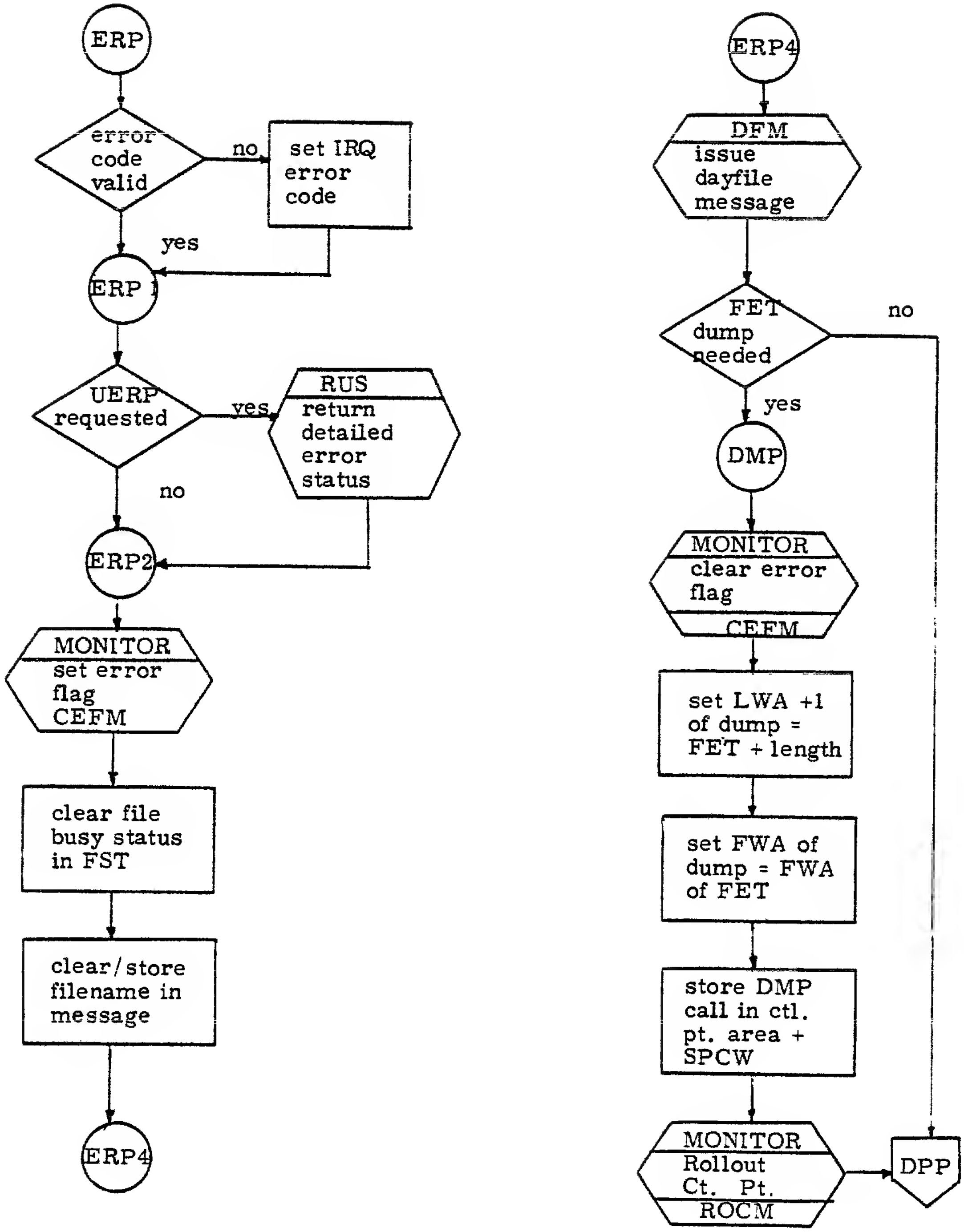


Figure 8-12. ERP - Error Processor (2CI)

8.3 2CA SUBROUTINES

Figures 8-13 and 8-14 are the flowcharts of the three subroutines in overlay 2CA. These are:

ISR - Identify Special Request
EVF - Evict Mass Storage File
EPF - Evict Permanent File

Table 8-5 TREQ is searched to map the request code in BS+4 into a function code stored in FC. The table contains the following entries.

TABLE 8-5. TREQ

Request Code	Function Code Name	Description
0100	OPE	OPEN, READ, NO REWIND
0104	OPE	OPEN, WRITE, NO REWIND
2110	OPE	Position multi-file set
0114	EVI	EVICT
0120	OPE	OPEN, ALTER, NO REWIND
0130	CLO	CLOSE, NO REWIND
0140	OPR	OPEN, READ, REWIND
0144	OPR	OPEN, WRITE, REWIND
0150	CLU	CLOSE, REWIND
0160	OPR	OPEN, ALTER, REWIND
0170	CLU	CLOSE, UNLOAD
0174	CLU	CLOSE, UNLOAD, RETURN
0300	OPE	OPEN, READ, NO REWIND
0330	CLO	CLOSE, NO REWIND
0340	OPR	OPEN, REWIND
0350	CLU	CLOSE, REWIND
0370	CLU	CLOSE, UNLOAD

8.4 2CB SUBROUTINES

Figures 8-15 through 8-20 are flowcharts of subroutines in overlay 2CB - Read Mass Storage. The following is a list of those subroutines; an asterisk indicates which ones are flowcharted.

*RMS - Read Mass Storage (Main Routine)
*LDB - Load CM Buffer
*WCB - Write Central Buffer
*EOF - Process EOF
*EOR - Process EOR
*CPR - Complete Read
CBS - Check Buffer Space
SBA - Set Buffer Addresses

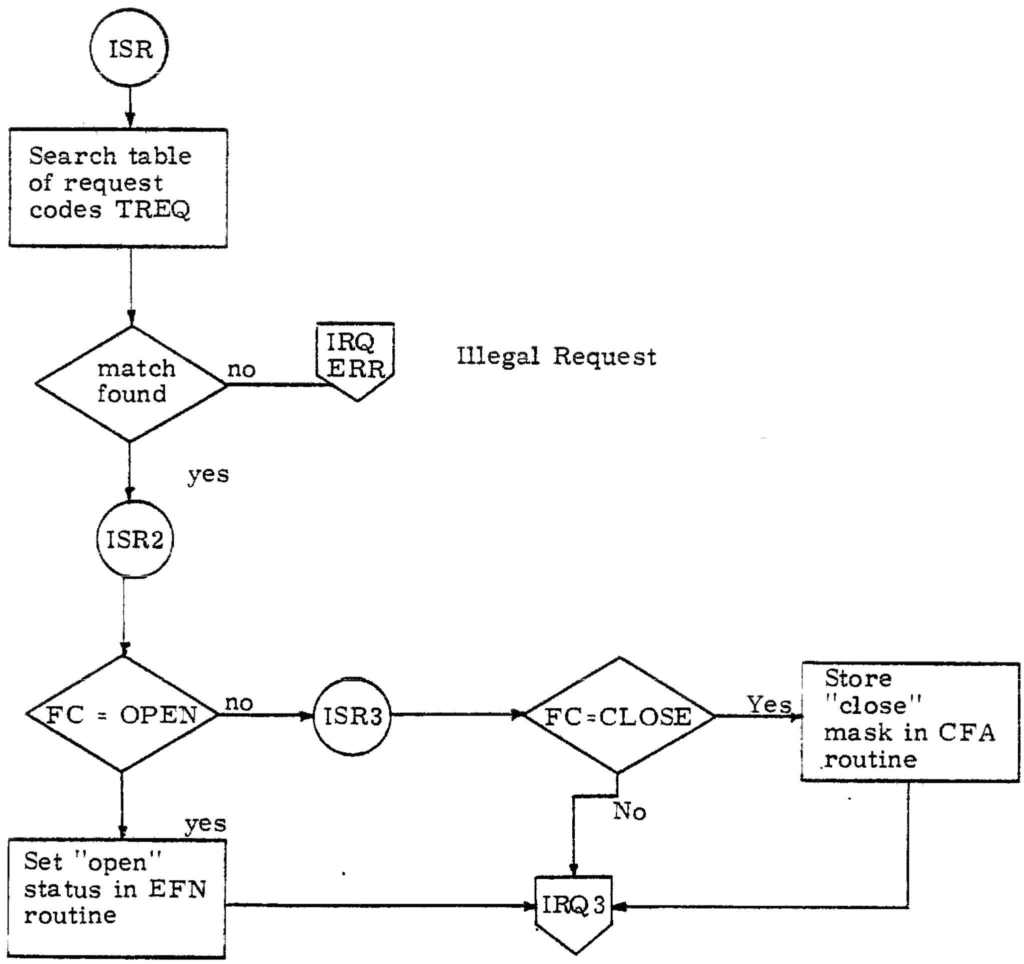


Figure 8-13. ISR - Identify Special Request (2CA)

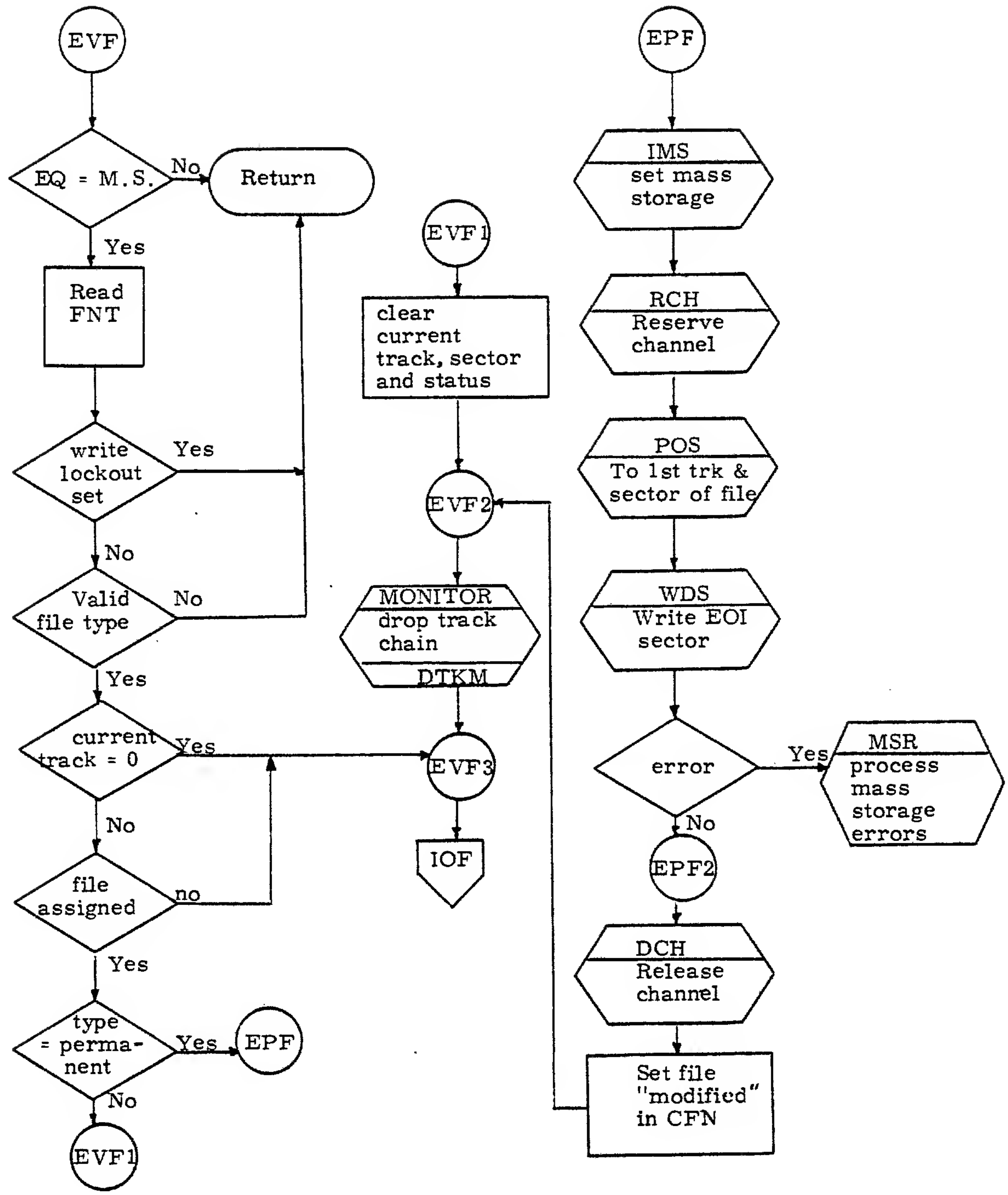


Figure 8-14. EVF/EPF - 2CA Subroutines to Evict a Mass Storage and Permanent Files

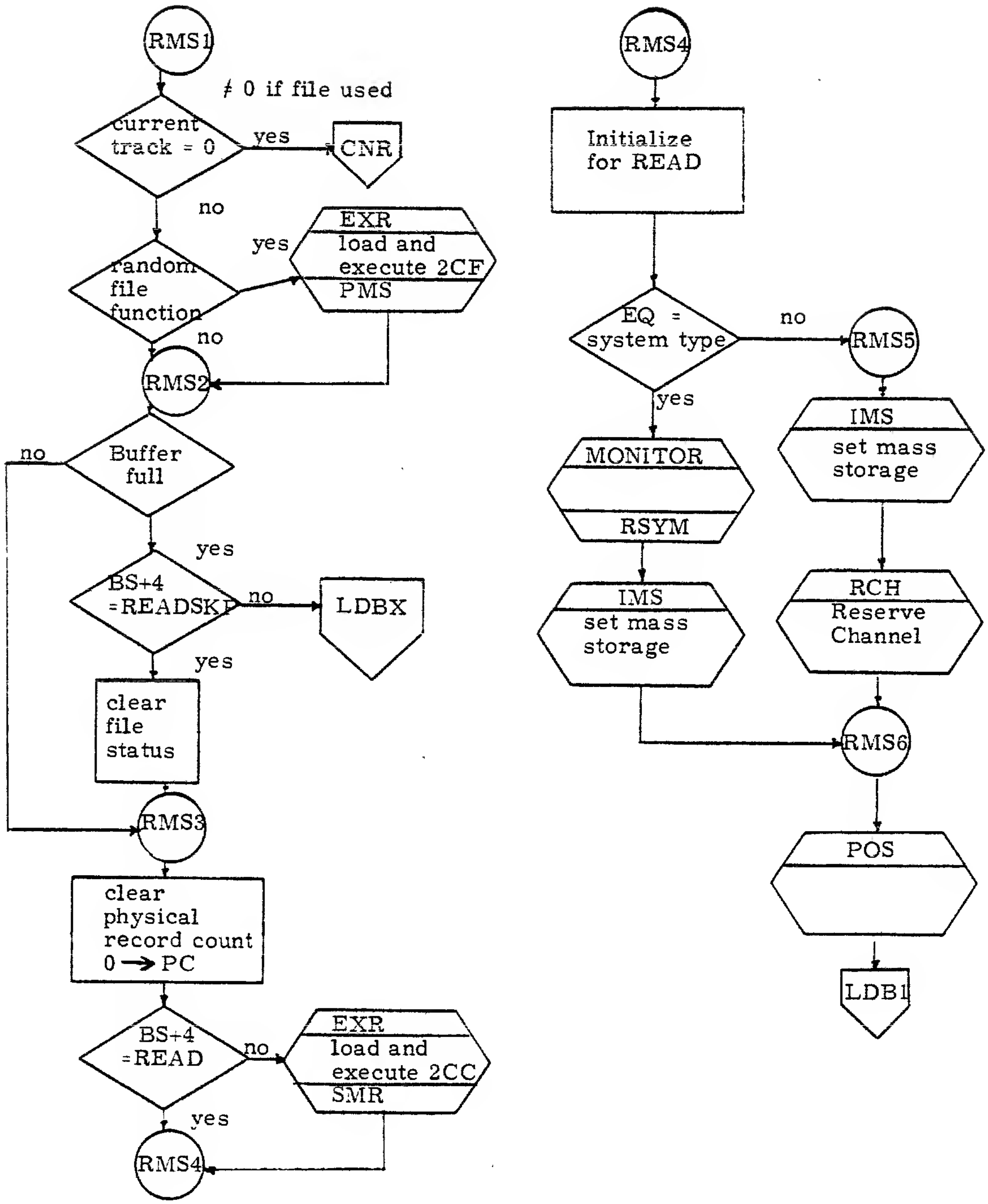


Figure 8-15. 2CB - Read Mass Storage

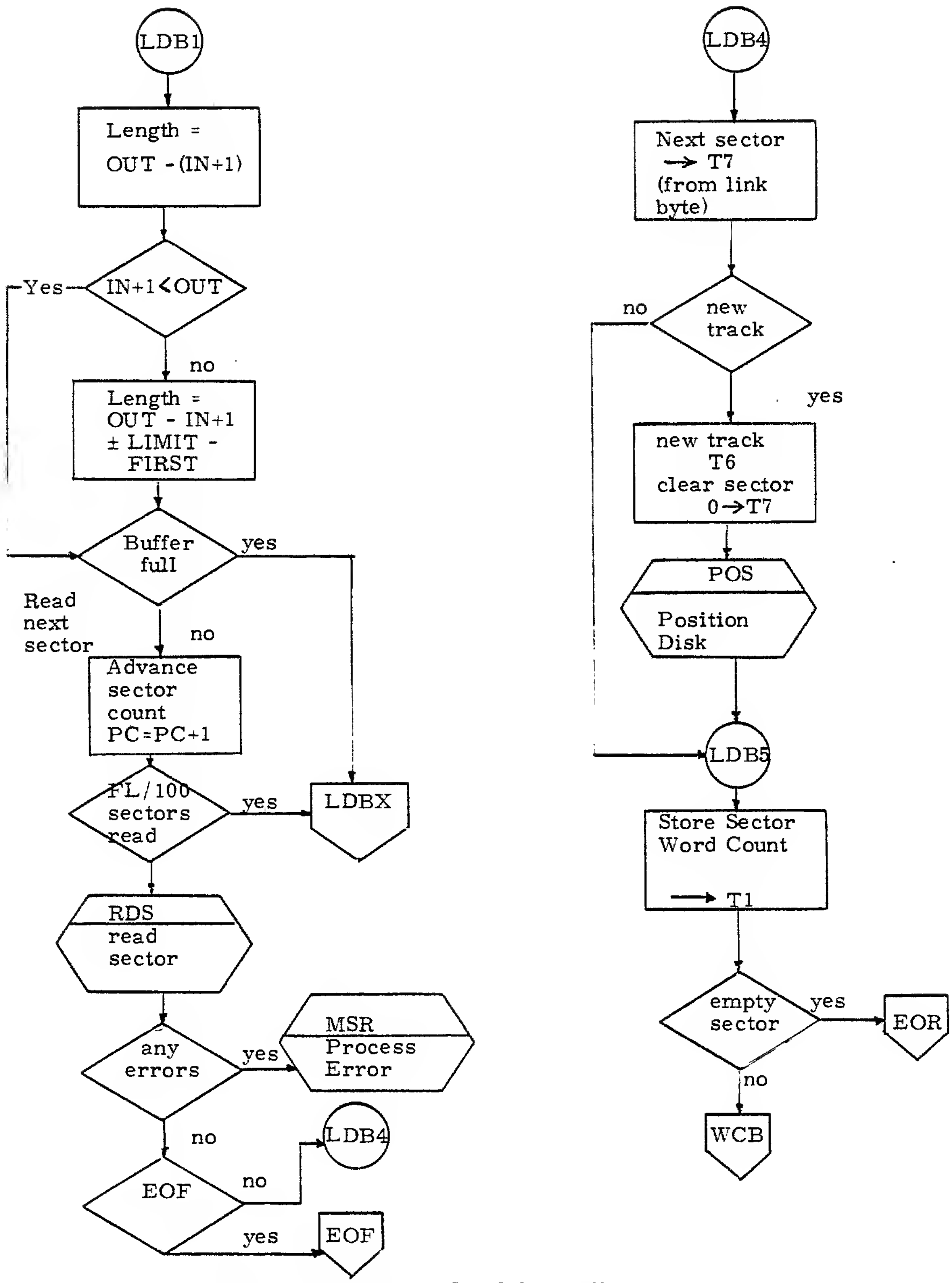


Figure 8-16. LDB - Load CM Buffer

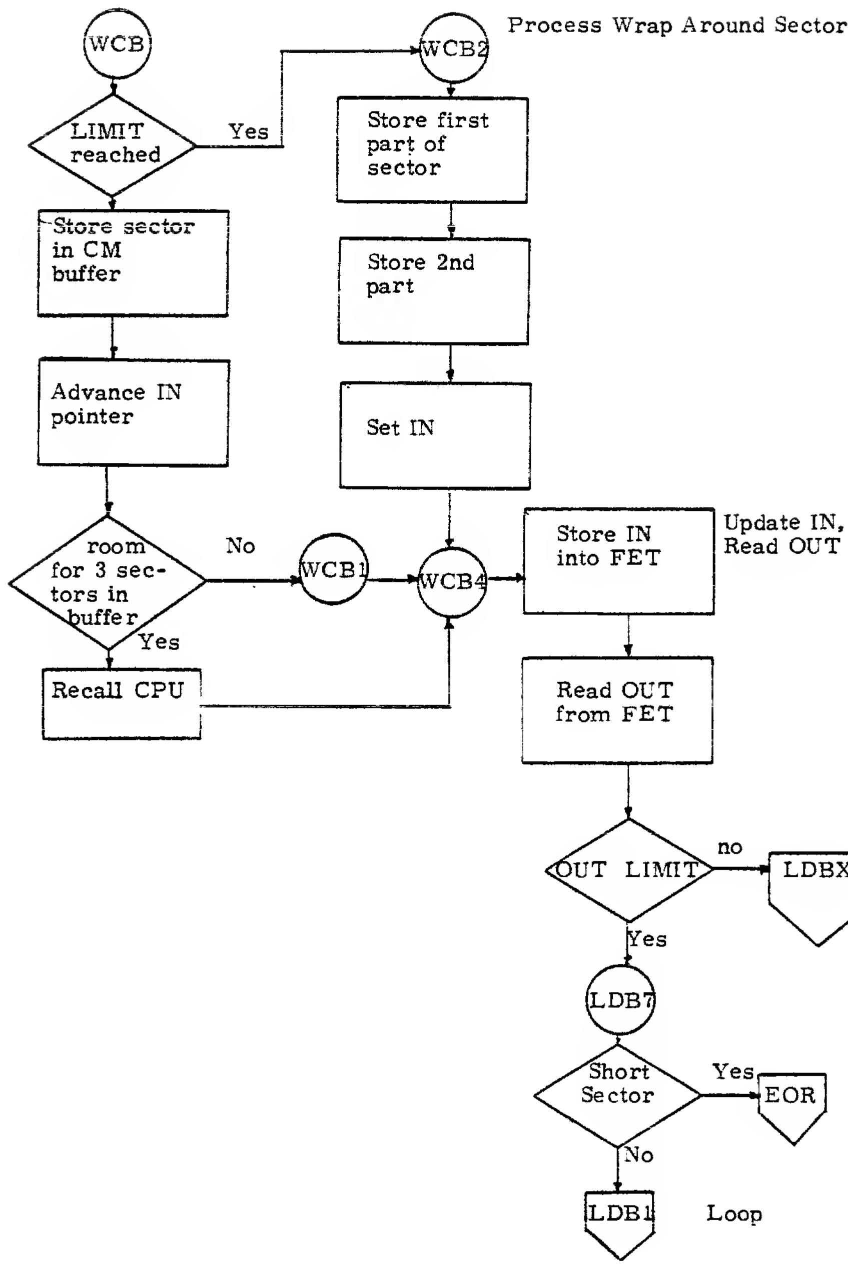


Figure 8-17. WCB - Write Central Buffer

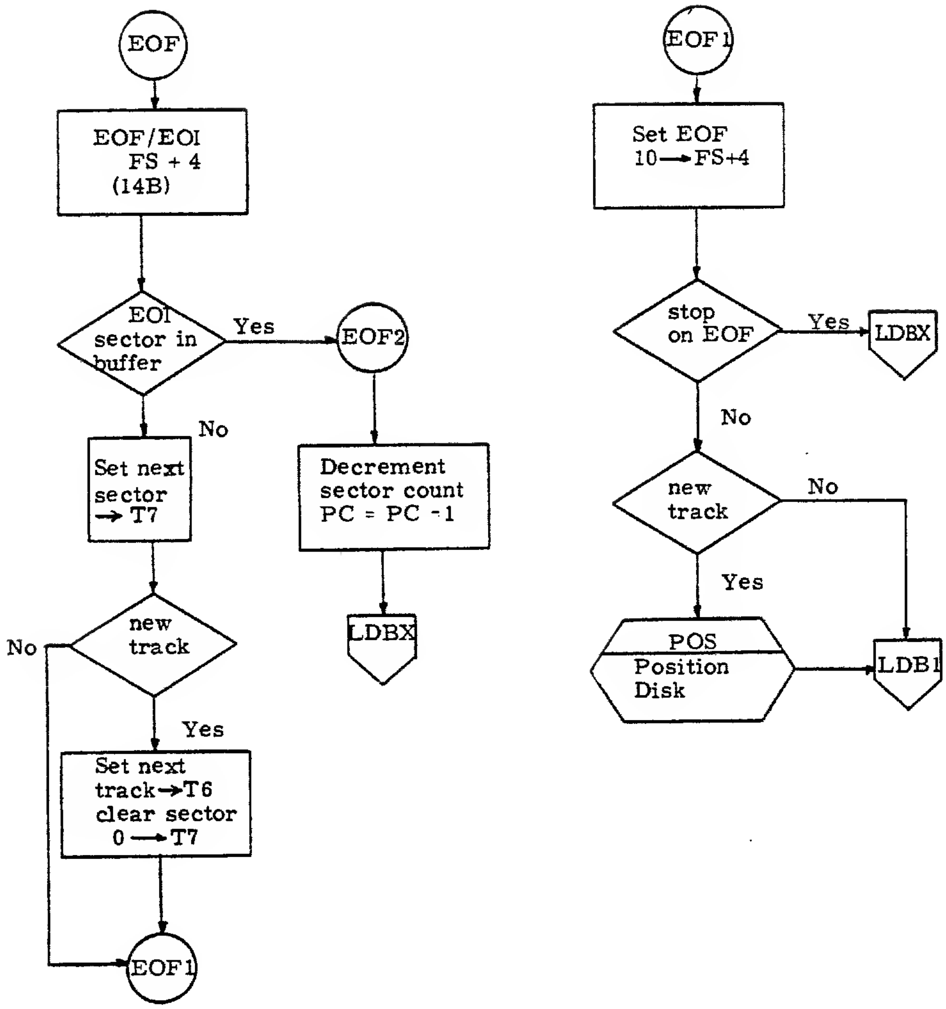


Figure 8-18. EOF - Process EOF

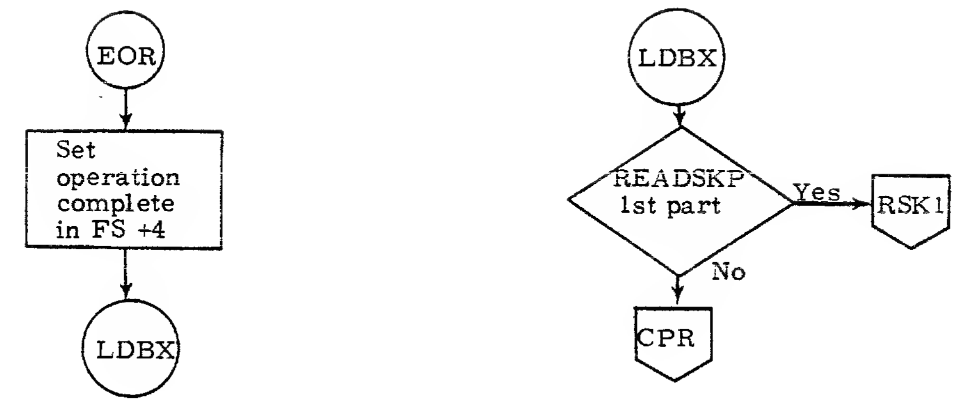


Figure 8-19. EOR - Process Error

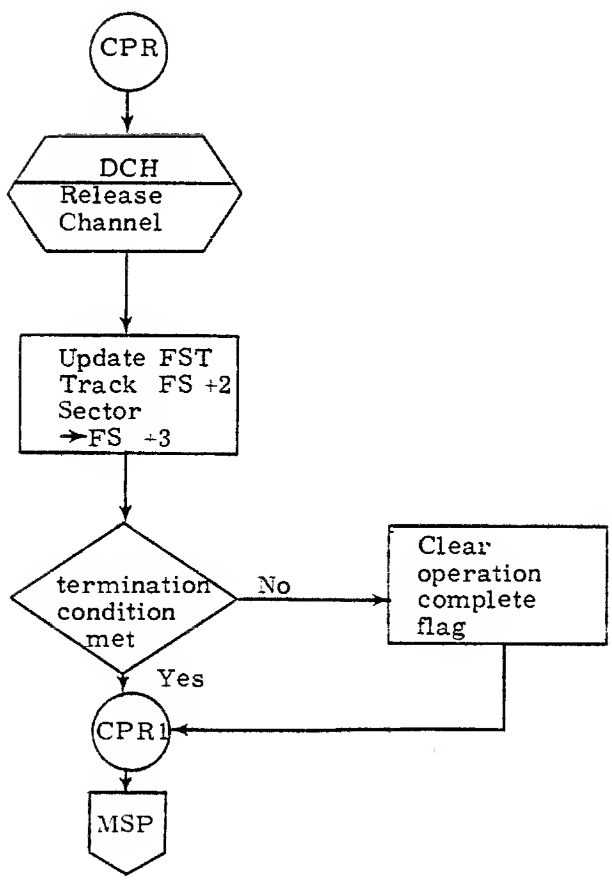


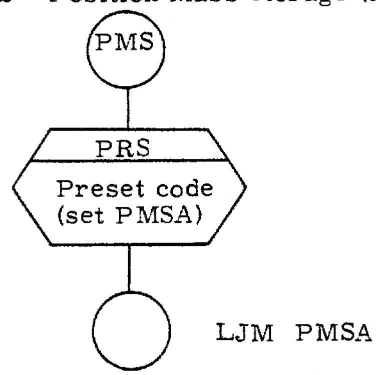
Figure 8-20. CPR - Complete Read

8.5 POSITION MASS STORAGE ROUTINE - PMS

Figure 8-21 is a partial flowchart of PMS. The Position Mass Storage routine is in overlay 2CF. PMS is called from three places in CIO:

1. Resident Processor PMS
2. RMS in 2CB
3. WMS in 2CD.

PMS - Position Mass Storage (2CF)



PMSA = RRD - Process Random Read
RWT - Random Write
SKF - Skip Forward
SKB - Skip Backward
BKS - Backspace
PMSX - Rewind
OPE - Open
CLO - Close

Function Processor Return

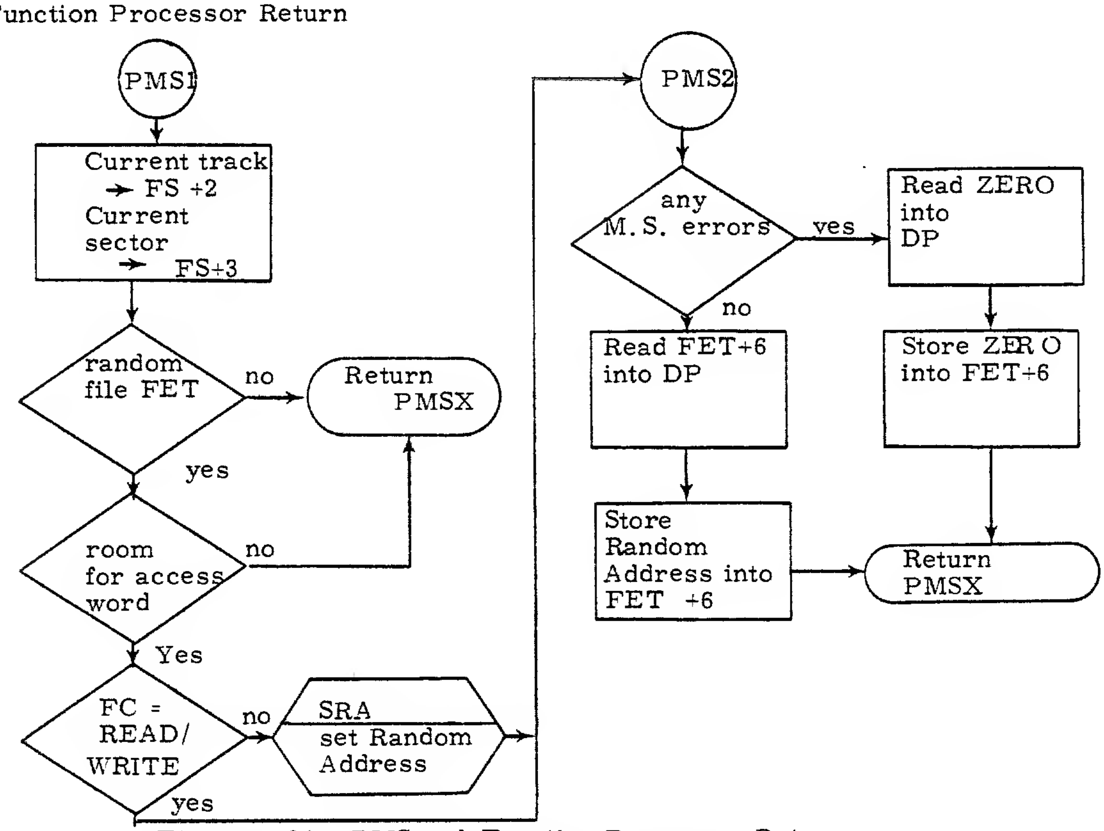


Figure 8-21. PMS and Function Processor Return

8.6 CIO TERMINATION ROUTINES

Figures 8-22 through 8-24 are flowcharts of the following CIO termination routines:

UFS - Update File Status

IOF - Set IN = OUT = FIRST

CFN - Complete Function

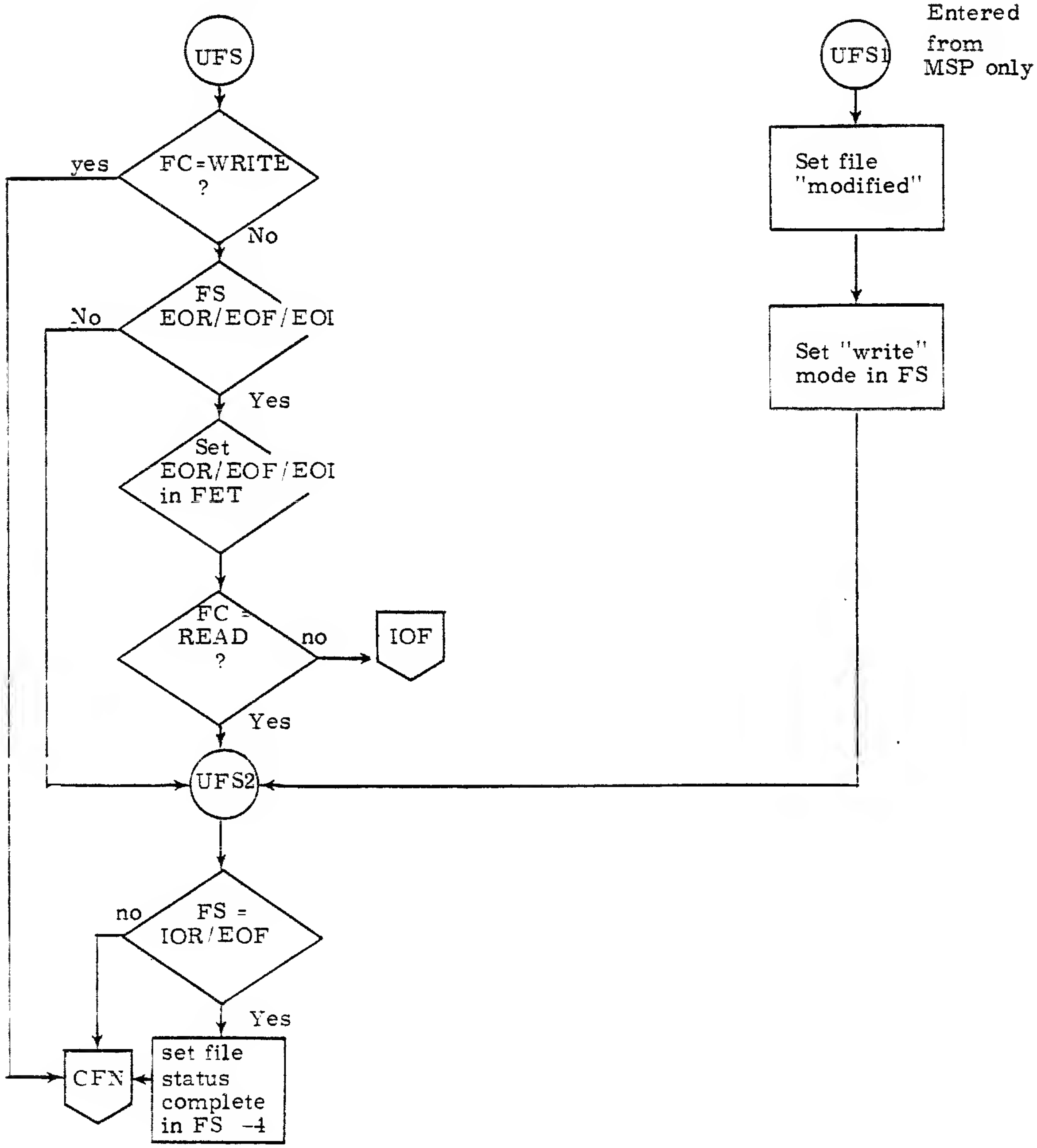


Figure 8-22. UFS - Update File Status

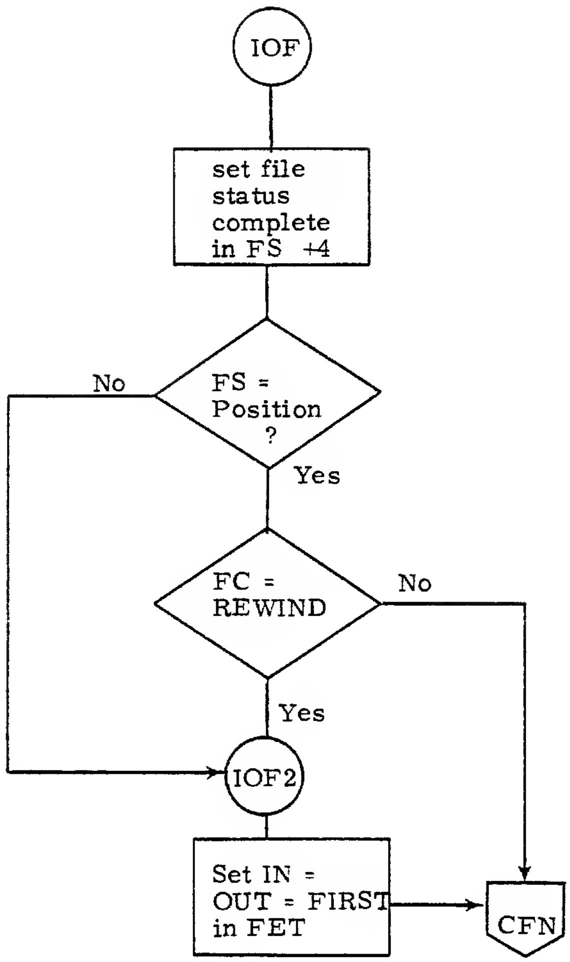


Figure 8-23. IOF - Set IN = OUT = FIRST

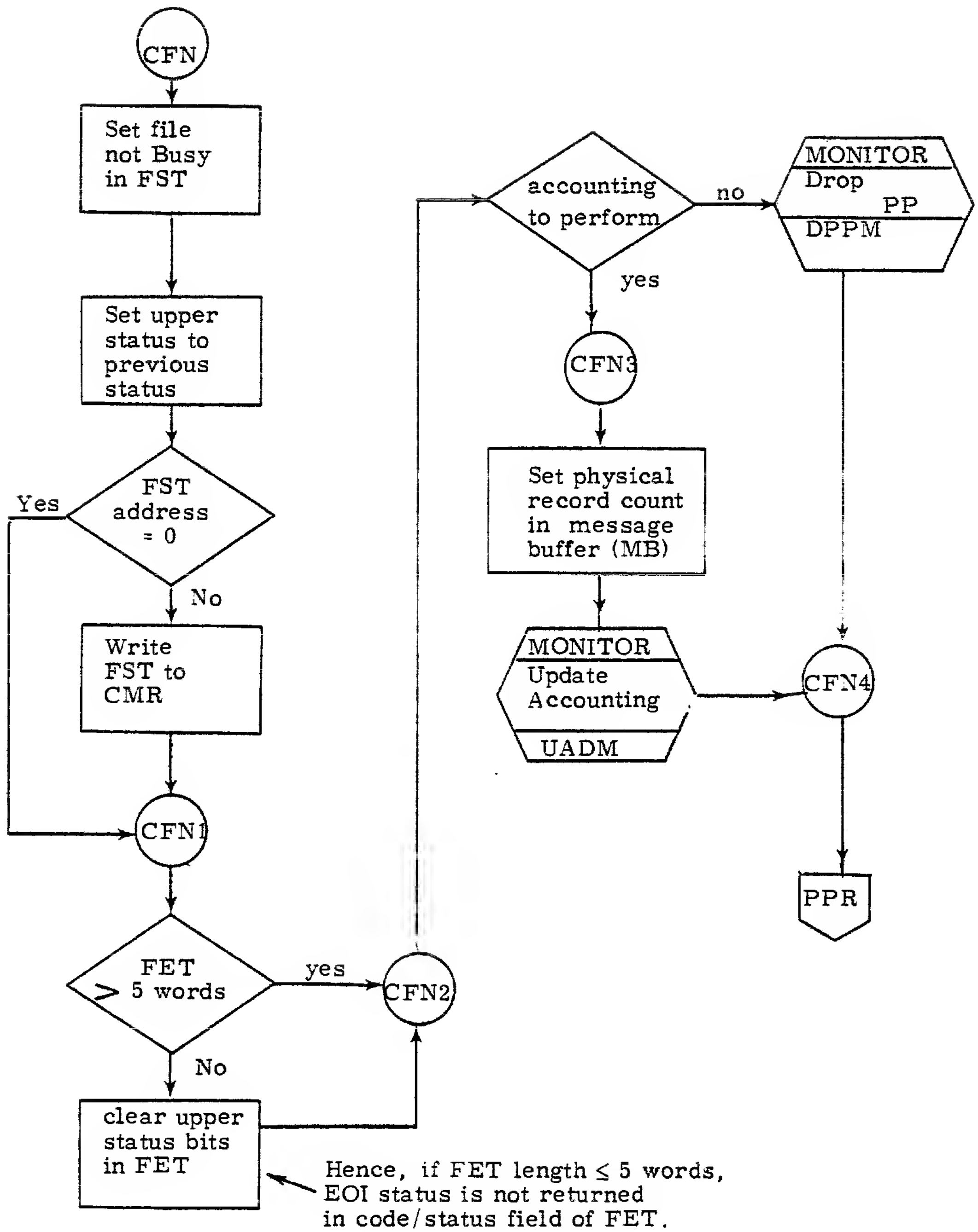


Figure 8-24. CFN - Complete Function

8.7 TERMINAL INPUT/OUTPUT ROUTINE - TIO.

Figure 8-25 is a flowchart of the terminal INPUT/OUTPUT routine, TIO. This routine is contained in overlay 2CG. TIO is only called from the PFN subroutine.

8.8 2CH SUBROUTINES

Figures 8-26 through 8-28 are flowcarts of the following three subroutines in overlay 2CH:

- PMT - Process Mag. Tape Operations
- MER - Mag. Tape Executive Request
- UDT - Unit Descriptor Table Read/Write

Basically, PMT sets up a 3-word parameter block and passes that information to MAGNET. The format of the three words is as follows:

Word 1 (FC, FC+1, FC+2, FC+3, FC+4) → MB

FC	FC+1	FC+2	FC+3	FC+4
Function code				

Word 2 (DP, DP+1, DP+2, DP+3, DP+4) → MB+1

Request code	FET options	level	ubc		mlrs
12	12	6	6	6	18

Word 3 (LM-3, LM-2, LM-1, LM, LM+1) → MB+2

Field Length		FIRST		LIMIT
12	6	18	6	18

where,

ubc = unused bit count } see FET +6 in K2.1
mlrs = max. logical record size } Reference Manual

Request code is from the FET. The upper bit is set if auto-recall was specified.

FET options are from byte 1 of FET+1 = EP, UP, xl.

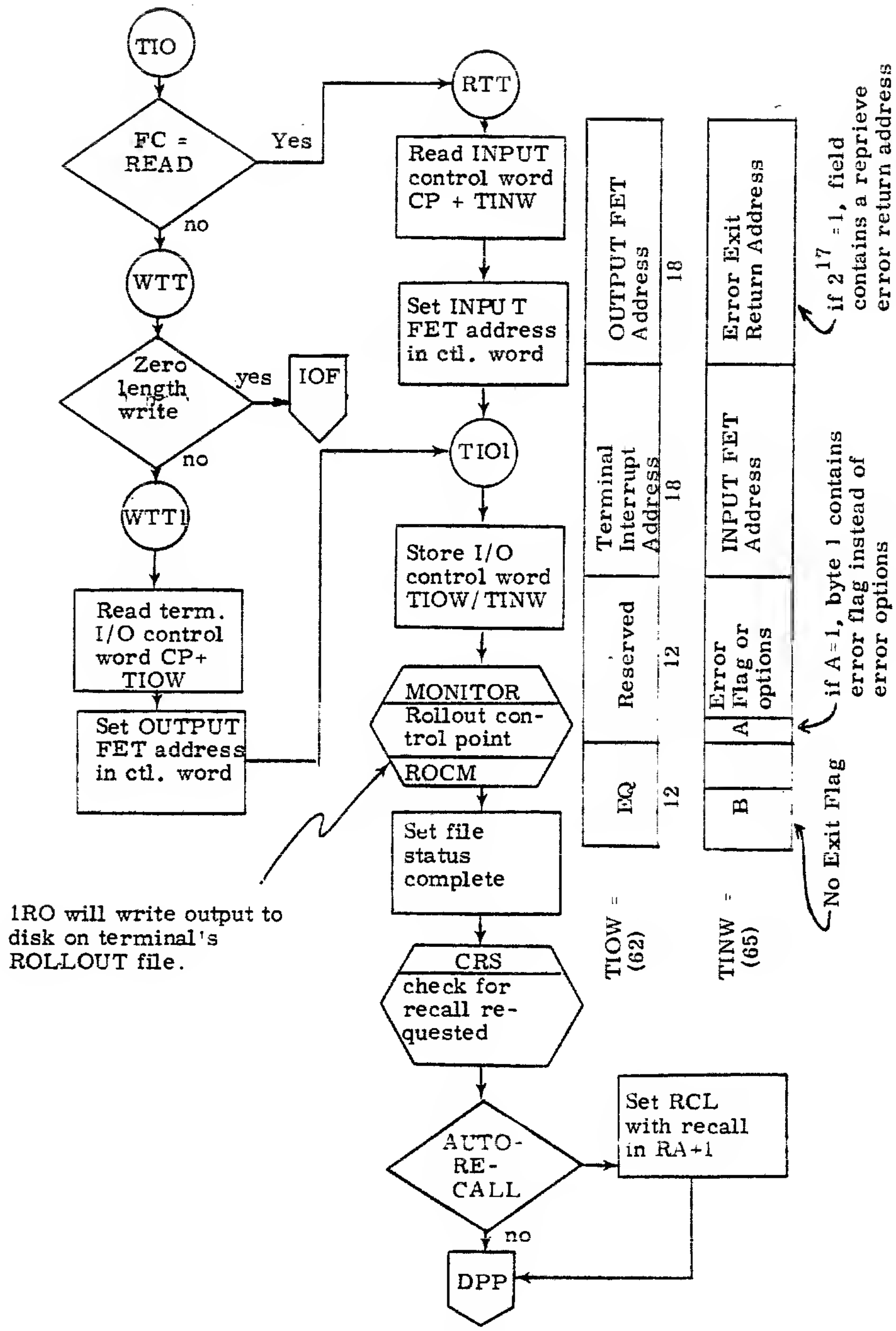


Figure 8-25. TIO Terminal Input/Output

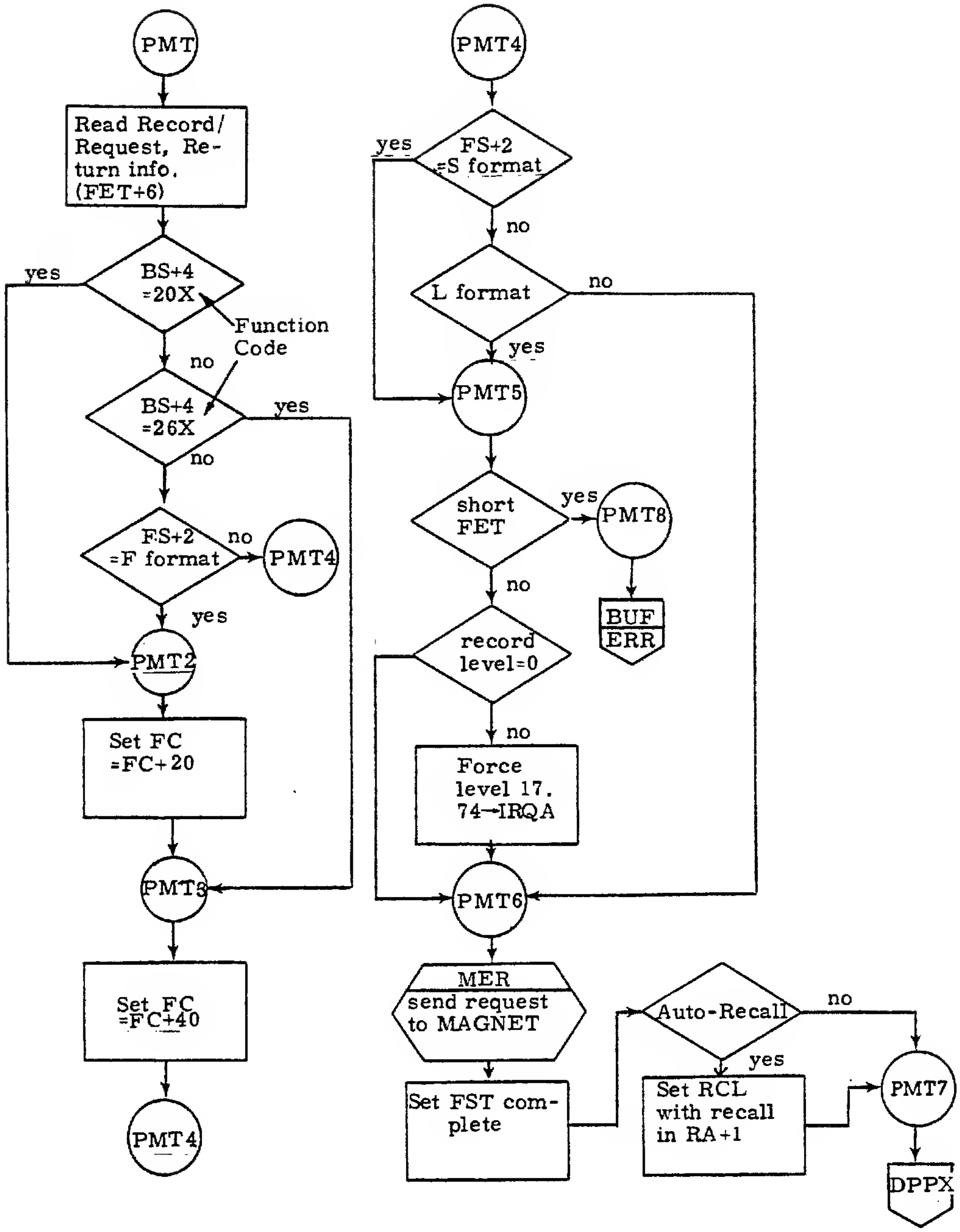


Figure 8-26. PMT - Magnetic Tape Operation

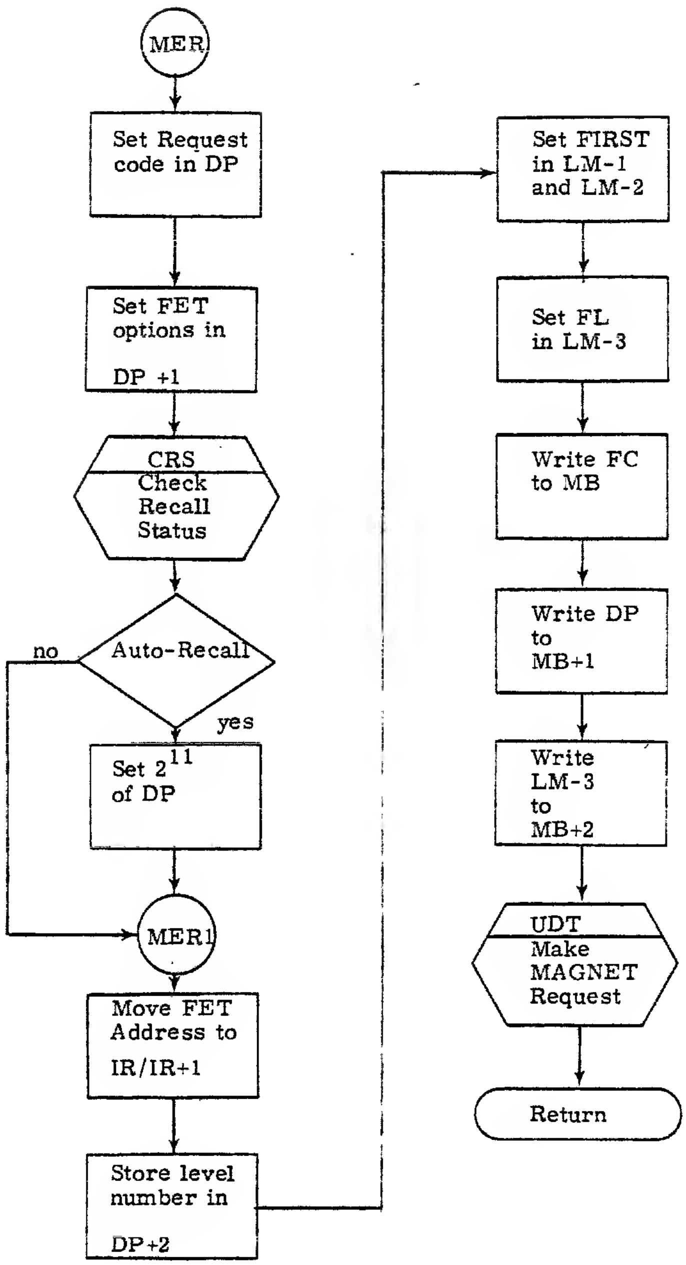


Figure 8-27. MER - Magnetic Tape Executive Request

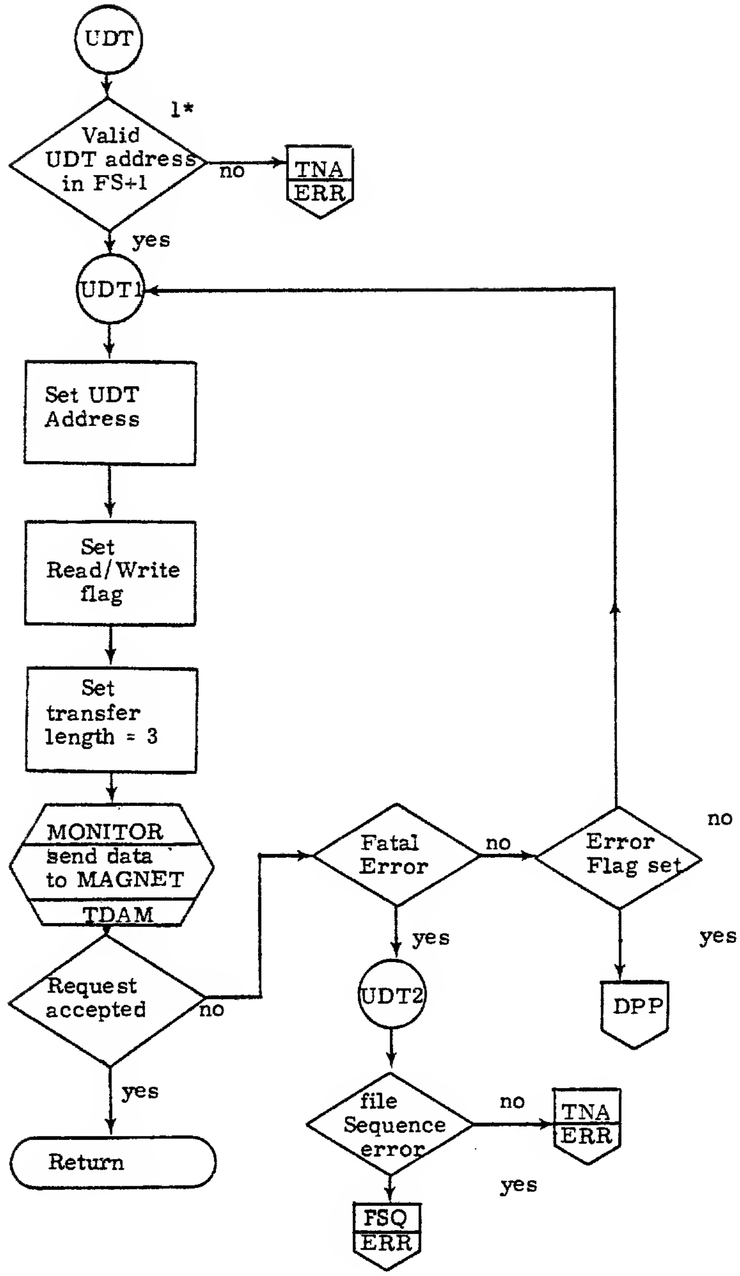


Figure 8-28. UDT - Unit Descriptor Table Read/Write

1* See UNIT DESCIPTOR TABLE in chapter 9.

RESOURCE CONTROL 9

9.0 INTRODUCTION

Resource control involves the allocation of the system magnetic tape and disk pack resources. The control of these resources is handled by the system routine (RESEX), while all magnetic tape operations are controlled by the magnetic tape executive, MAGNET. This section describes these two executives.

For a description of magnetic tape formats, consult the KRONOS 2.1 Reference Manual, Section 9. Also, Section 5 of the manual contains the control card call, RESOURC, for initiating the resource executive, RESEX.

Figure 9-1 shows an overview of the system routines involved with resource control and allocation.

The whole concept of MAGNET-RESEX is to allow overcommitment of tapes and removable packs.

MAGNET runs at a CP and is a repository of information for RESEX and the System. The E,P display is updated by RESEX and displayed by DSD. The E, T display is the UDTs and is updated by MAGNET and displayed by DSD. DSD commands from the console are placed in MAGNET and updated by MAGNET. CIO places tape read/write requests in the UDTS.

MAGNETs main duty is to receive requests from DSD and RESEX and to initiate 1MT to process CIO requests.

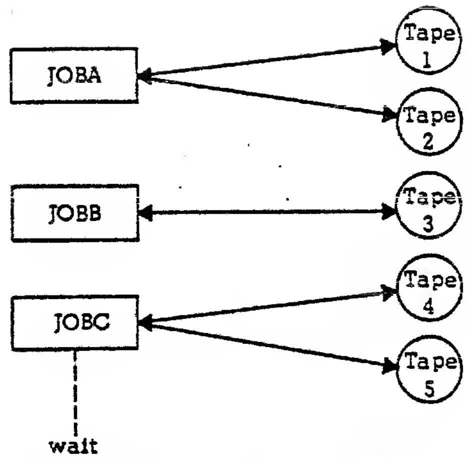
RESEX is loaded at a users CP, and the user is saved on a DM* file is necessary. RESEX determines the users requirements vs system availability and user validation. Overcommitment is exercised if necessary. In order for RESEX to determine recource availability and overcommitment, all user demand information is save on a Fast Attach PF called RESEXDF, all resource activity information is save on a fast attach PF called RESEXVF.

In the old days of KRONOS 2.0, tape scheduling was based on tape unit availability.
For example: assume 5 tape drives.

JOB A needs MT2

JOB B needs MT1

and JOB C needs MT3



JOBC would wait for JOBA or JOBB to release a tape so it could continue.

Now with the advent of KRONOS 2.1, the system can overcome deadlock situations.

Some definitions are:

Deadlock - Two or more unsatisfied tape jobs have tied up all the units in the installation.

Potential Deadlock - Two or more unsatisfied tape jobs have been assigned tapes in such a manner that there are not enough free units remaining to satisfy the maximum requirements of any of them.

9.0.1 Deadlock Condition

Assume 8 tape drives.

	MAX	UNFILLED	ASSIGNED
JOBA	3	1	2
JOBB	3	1	2
JOBC	5	1	4

None of these jobs is in a position to release a drive. Hence the Deadlock.

9.0.2 Deadlock Prevention

	MAX	UNFILLED	ASSIGNED
JOBA	3	1	2
JOBB	3	1	2
JOBC	5	1	3

JOBC requests a tape - Refused since it could cause a Deadlock condition.

JOBB or JOBA requests a tape - Granted since it would not cause a Deadlock condition.

9.0.3 Overcommitment

Many jobs with tape drives can be scheduled and drives assigned as long as a potential deadlock can not occur. As long as at least one job can complete, tapes can be scheduled. When at least one job will be unable to complete due to a tape drive assignment, that assignment is deferred (i.e. not allowed, even if the operator assigns it).

9.0.4 Tape Scheduling Deadlock Prevention Algorithm

The system is "safe" if there exists at least one active job such that:

a. There are enough currently unassigned tape units to satisfy the maximum requirements of the job and

b. When this job completes, it will release enough tape drives such that the total number of drives then available are sufficient to satisfy the maximum tape requirements of at least one other job such that:

9.0.5 Tape Assignment Dynamic Tape Unit Status Checking

Periodic checking for ready or not ready status.

- E, T always current!

Advantages:

- Automatic assignment can occur at any time.
- Automatic a signment of unlabelled tapes.
- Improved reel swapping.

Tape Assignment Objectives

- Improve management of tape units
- Increase automation of tape assignments
- Flexibility in assignments
- Assist operator
- Basis for improvements

Tape Assignment New Features

- Automatic tape assignment by VSN
- VSN control card
- Tape job prescheduling display
- Tape drive overcommitment
- Dynamic unit status checking

9.0.6 Tape Preassignment Display

Resource Mounting Preview

NO	EQ	PN/VSN	USERNO	RING
15	MT	TAPE1	MLO	I
31	MT	A216B	ABJ	
42	DI3	CATCH	FISH	

The following examples will attempt to clarify the KRONOS 2.1 philosophy on OVERCOMMITMENT of tape/private pack equipment.

Example 1 shows how deadlock can occur.

Example 2 shows the classic textbook case of overcommitment.

Example 3 shows a typical overcommitment.

Example 1

Example of DEADLOCK (without RESOURCE protection)

Assume 4 tapes

	Job1	Job2	MTs Available
	RESOURC(MT = 2)	RESOURC(MT=2)	4
	REQUEST(F1,MT)		3
		REQUEST(F1,MT)	2
	REQUEST(F2,MT)		1
		REQUEST(F2,MT)	0
1.	RESOURC(MT=3)		
2.	REQUEST(F3,MT)		-1
3.		RESOURC(MT=3)	
4.		REQUEST(F3,MT)	-2 DEADLOCK

1. If Job1 is not aborted then:
2. Job1 rolled out since no tapes are left to assign.
3. Job2 rolled out since no tapes are left to assign.
4. Neither job can complete and neither job can release any tapes, so both jobs and the tape system are locked up tight==DEADLOCKED.

   Hence, at point 1. Job1 is aborted since the resources desired are not currently available. However, Job2 is not aborted, since when Job1 was aborted it released its two tapes. So, Job2 resources are available.

Example 2

Assume System has 4 MTS Available

Job 1	Job 2	MTS Available
JOB 1	JOB 2	4
ACCOUNT	ACCOUNT	4
RESOURC (MT=3)	RESOURC (MT=3)	4
REQUEST (F1, MT)*		
REQUEST (F2, MT)*		2
	REQUEST (F1, MT)*	1
	REQUEST (F2,MT)—RO	1
REQUEST (F3, MT)*		0
RETURN (F2)* decrenment demand count by 1.0		1
	REQUEST (F2,MT)*—RI	0
END - OF - JOB		2
	END - OF - JOB	4

* REQUEST PROCESSED WHEN ENCOUNTERED

Note that UNLOAD will return a tape without decrementing the demand count.

If this Job 1 attempts another request, it will be aborted.

Example 3

Assume System has 3 MTs Available

JOB1	JOB2	JOB3	MTs available
JOB1.	JOB2.	JOB3.	
ACCOUNT,,.	ACCOUNT,,.	ACCOUNT,,.	3
RESOURC(MT=3)*	RESOURC(MT=1)	RESOURC(MT=2)*	3
REQUEST(F1,MT)*			2
REQUEST(F2,MT)*			1
		REQUEST(F1,MT)RO	1
	REQUEST(F1,MT)*	ROLLED OUT	0
REQUEST(F3,MT)RO			
ROLLED OUT			
	RETURN(F1)*		1
RI			
REQUEST(F3,MT)*			0
ROLLED IN			
	END-OF-JOB		
		RI	1
RETURN(F2)*		REQUEST(F1,MT)*	0
decrements demand count by 1.		ROLLED IN	
REQUEST(F4,MT) ABORT "DEMAND EXVEEDED" END-OF-JOB			2
		REQUEST(F2,MT)*	1
		END-OF-JOB	

*REQUEST PROCESSED WHEN ENCOUNTERED

The following synopsis shows the interaction for Tape Scheduling

1. without VSN
2. with VSN
3. private pack scheduling.

Tape Scheduling Without VSN

User	System	Operator
1. Attempt to access file on tape via REQUEST LABEL.		
	2. RESEX loaded and entered at entry point LABEL or REQUEST (as a CC call, i.e. not an SPCW call).	
	3. RESEX uses the macro which makes an RA + 1 request for LFM.	
	4. LFM issues the REQUEST B-display message and waits for the operator to assign equipment. The control point may be rolled out.	
		5. Operator scans E,T display for a free drive, mounts the tape and assigns the equipment to the control point.
	6. LFM reads equipment from CPA, OAEW word, and saves it in an FNT/FST entry. Then LFM calls RESEX via SPCW word in CPA and drops. (Original RESEX is saved on DM* file). The equipment number is passed in the Status field of the SPCW.	

Tape Scheduling Without VSN (Continued)

User	System	Operator
	7. RESEX reads the UDT entry associated with the eq in MAGNET's fl (via RSB), sets up a request block and sends it to MAGNET (RCAL block via SIC).	
	8. MAGNET assigns the UDT to the job and completes the request. (sets up the rest of the UDT entries).	
	9. RESEX reads the UDT entry again and when complete, calls LFM to complete the FNT/FST with the UDT address, updates the E, P (preview buffer) and ends.	
	10. The original REXEX is rolled back in from DM*, updates the E,P and ends.	
	11. Control point is advanced.	

Tape Scheduling With VSN

User	System	Operator
1. Attempt to access file on tape with REQUEST (VSN=...) LABEL (VSN=...) VSN		
	2. RESEX is loaded as a CC call and is entered at appropriate entry point.	
	3. Parameters are processed and LFM is called to create FNT/FST entry with eq type TE if not already present.	

Tape Scheduling With VSN (Continued)

User	System	Operator
	4. RESEX reads all UDTs, via RSB, and looks for a match on VSN.	
	5a. If duplicate USN's are found LFM is called to have operator assignment.	
		6a. Operator scans E,T and assigns tape.
	5b. If VSN is not found, RESEX sends preview information to MAGNET and calls LFM to enter Timed/Event rollout (the time interval is 2 minutes and the event is a folded checksum of the VSN).	
		6b. Operator scans E,P and mounts tape with proper VSN.
	5c. RESEX finds single VSN.	
		6c. No operator intervention necessary.
	7. RESEX sets up a request block and sends it to MAGNET (SIC to RCAL).	
	8. MAGNET assigns the UDT to the job and completes the request (sets up the rest of the UDT entries).	
	9. RESEX reads up the UDT's and finds the tape assigned to a UDT and calls LFM.	
	10. LFM completes the FNT/FST and changes assignment from TE to proper MT or NT, (est ord) and ends.	
	11. RESEX updates the E,P display (Preview Buffer) and ends.	
	12. Control Point advances.	

Automatic Scheduling of Auxiliary Packs

User	System	Operator
1. Attempt to access file on aux. device ATTACH (A/PN=PACK,NA)		
	2. PFM called and detects a. "PACK" not present b. User requests wait (NA)	(NO ABORT)
	3. PFM requests RESEX processing	
	4. RESEX after determining if request can be honored enters data in Preview buffer (MAGNET) and rolls out for 2 minutes.	
		5. Operator scans E, P display and notes name and type of pack required.
		6. Operator scans E, M display and then mounts pack on available spindle (must be removable equipment)
	7. Every 1 minute CMS will status drives - if "PACK" is available it will be "recovered" and set as available in MST.	
	8. RESEX rolled out, will roll in every 2 minutes and interrogate to see if "PACK" is available.	
	9. If it is, resource files updated and PFM is recalled to process ATTACH function and control then returned to user.	

Automatic Scheduling of Auxiliary Packs (Continued)

User	System	Operator
10. User RETURNS file A		
	11. System updates appropriate tables, MST to indicate device is no longer being used (user count).	
		12. Operator via E, M display notes that PACK has no active use.
		13. Operator enters UNLOAD. This prevents further PF requests to PACK.
		14. Operator removes pack.
	15. CMS on next cycle recognizes that device is not ready and "clears" MST accordinaly.	

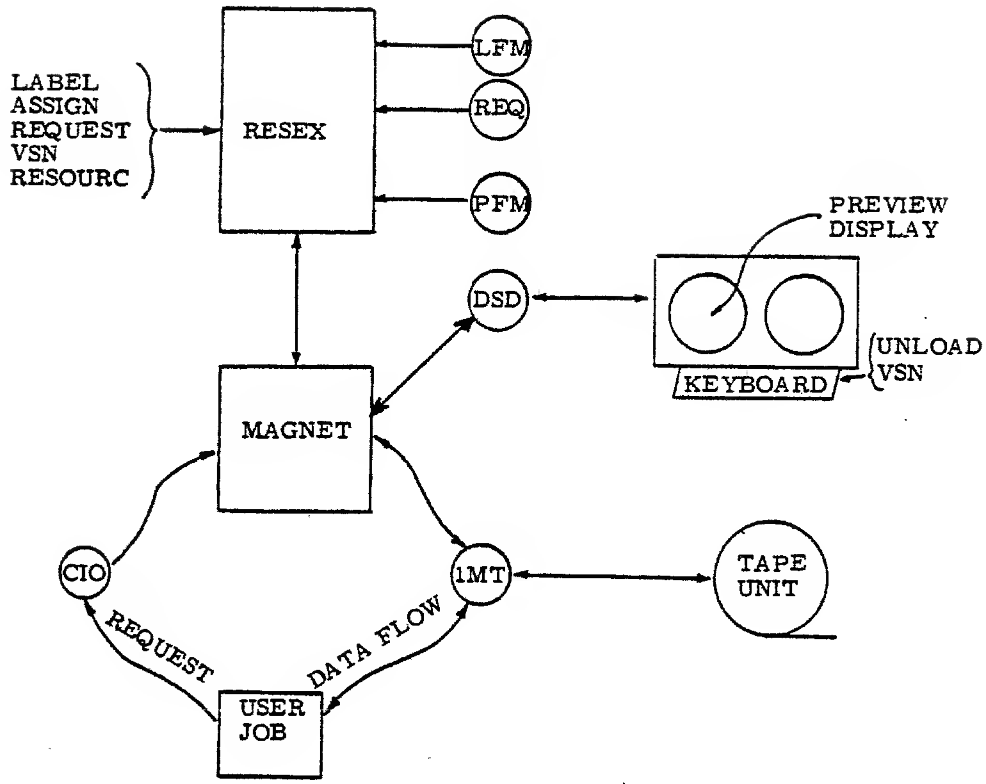


Figure 9-1. RESEX/MAGNET System Interface

3MK	-	READ error processor
3ML	-	WRITE function processor
3MM	-	Write long block processor
3MN	-	Coded WRITE processor
3MO	-	Write label processor
3MP	-	WRITE error processor
1LT	-	Long block processor

1MT overlays itself extensively to conserve space. It uses areas in PP Resident and the 5-byte header on PP routines, therefore due care must be taken when attempting modifications.

All magnetic tape equivalences are defined in the common deck, COMSMTX. These equivalences are used by MAGNET, RESEX, 1MT, CIO, DSD, 0RF, and 1DS.

9.1.1 MAGNET Control Point Initialization/Termination

The control point for MAGNET is initialized in the same manner as TELEX or TRANEX. That is, DSD calls 1DS to process the operator type-in, MAGNET. 1DS then calls 1MT to initialize the executive. 1MT determines that this is an initial call to MAGNET and executes overlay 3MA to perform control point initialization. (1MT determines that this is an initial call by checking for "MA" in word JNMW of the control point area). 3MA performs the following operations:

1. Calls INI to store jobname of "MAGNET" in control point area with system origin type (SYOT) set. Calls monitor to set priority =76.

2. Calls RQS to request field length.

3. Calls SCC to set up the control card buffer as follows:

   MAGNET
   MAGNET1
   EXIT.
   MAGNET1.

4. Calls EST to preprocess EST entries. A list of tape channels and equipments is created in PP memory.

5. Calls PCC to call 1AJ to process the first control card. (MAGNET loaded by 1AJ.)

6. MAGNET is loaded in the CP.

7. Calls BDW to build an equipment definition list in MAGNET's FL at RA+ UINT=7700B. Only used during initialization to build UDTs, following PRESET code in MAGNET.

8. Calls BIW to build interlock words in MAGNET's FL at RA+UITW (RA+10B). These words are used by MAGNET to call and interlock with 1MT. (An entry is passed to 1MT via a TLX call in RA+1.)

9. Sets an inter-control point word in CP area as follows:

59 — 48	47 — 30	29 — 18	17 — 0
RCALL	RCAL	PBUFL	PBUF
12	18	12	18

where:
- RCALL - Length of RCAL (10B)
- RCAL - RESEX request block buffer
- PBUFL - Length of PBUF
- PBUF - FWA of preview buffer (read by DSD to build PREVIEW display)

10. Drops 1MT. MAGNET is now in control.

9.1.2 MAGNET Initialization

After Step 6 of paragraph 9.1.1 is completed, 1AJ loads the CP portion of the executive MAGNET, and execution begins at the preset routine, PRS. PRS clears the interface area from UITW+1 through TRPO (RA+11B - RA +140B). The interlock word UITW is cleared and MAGNET waits until step 7 is nearly complete before continuing processing. That is, MAGNET waits for 1MT to build the equipment definition list. PRS then calls PEQ and REL before jumping into the main control loop. PEQ builds a list of UDTs (one for each unit as sensed by 1MT). REL performs the required instruction modification in the main routine where the OPDEF's have been used. The UDTs start at TDTAB and overlay the preset code. A maximum of 16D (MUNIT in COMSMTX) UDTs are established in MAGNET's FL. PEQ also sets up a pointer word in RA+3, called UBUF, which points to the list of UDT entries. UBUF has the following format:

		Number of Units	LWA of UDT	FWA of UDT
RA+3	UBUF	Number of Units	LWA of UDT	FWA of UDT
		12	24	24

where: FWA of UDT = TDTAB = 767B currently
LWA of UDT - Dependent on number of units

Each UDT entry is UNITL words long (currently, UNITL = 20D). PEQ sets the SED function (Set Equipment Definition) in each UDT entry, therefore 1MT will be called to determine the type of each unit.

PEQ sets up another low core pointer, UQUE. UQUE (RA+4) specifies the first word address of the queue table. This table follows the UDT list, and is initialized with 10B empty entries. The queue table is terminated with two words of all 7's. The Last Word Address (LWA) of the queue table is stored in FLSW+1, currently RA+30B. FLSW contains the starting FL from A0. Figure 9-2 is provided to show the memory map of MAGNET after initialization. The PREVIEW buffer (PBUF) is built by RESEX and transmitted to MAGNET where DSD obtains information to be displayed.

Address	Bits 59–48	Bits 47–24	Bits 23–0	Label / Notes
RA + 0				
1				
2				
3	# of entries	LWA of UDT	FWA of UDT	UBUF
4			FWA of Queue	UQUE
5	unit swap flow			USWP
6–7	Two word DSD request buffer			XREQ
10 ...	1 MT interlock words			UITW
20	4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			Terminated by Two words of bit 59 set
21				
22	Field Length status word			FLSW
23	1 MT function codes			TFUN
34	4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			terminator
35–44	RESEX request block Buffer used by RESEX to request UDT assignment to Tape			RCAL
45–143	E, P Preview buffer data			PBUF
144	Code			TPRO
1060	UDT, 22D words per UDT 1 UDT per Tape drive			TOTAB
	4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			terminator
				Queue table 10 empty entries initially
	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7			
	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7			two word of all 7 terminates table

Figure 9-2. MAGNET Memory Map (Level 5)

address for 1st UDT in example

Address	Name	Fields (bits 59 · 47 · 35 · 23 …)							
1360	UXRQ	RS	FUNCTION	MODE(MD)	PA	PB			
1361	UCIA	ICIO	SKIP COUNT	FET ADDRESS					3 word block sent by CIC
1362	UCIB	R ECIO	FET OPS	LNU		Record Request Return			
1363	UCIC	FL	FIRST	LIMIT					
1364	UST1	ED	HP	EC	E	DS			
1365	UST2	Disk PRUs	BLOCK count	User OPS					
1366	UST3	Last GOOD Record	Error Parameter	Den	CV				
1367	UST4	WC	OV	UBC	FORMAT	EST	NB	SP	
1370	UST5	MTS detailed Status							66X words
1371	UST6	MTS Status continued	MTS Format						
1372	ULRQ	MAGNET Last Request							
1373	UREQ	20NN	ADD	B2	B3	X5			
1374	UFLA	MAGNET flags							
1375	UFSQ	Job Seq Num	CP NUM	SIND	VSN Rand Index				
1376	UJBN	Jobname	OT						
1377	UUFN	User Number	FAM	ESW	Accv				
1400	USVN	Volumn Serial Number (VSN)							
1401	UFID	FILE Identifier							
1402	UFSD	FILE Id continued	File Section Num						
1403	USID	SET Id	Acch	File Seq Num					
1404	VGNU		GEN VER	GEN Num					
1405	UDAT	Creation Date	Expiration Date						

22D = 26B words per entry

Description contained in COMSMTX.
A partial listing follows.

Figure 9-3. Unit Descriptor Table (UDT) Entry (Level 5)

```
          *          ASSEMBLY CONSTANTS.

   20     MUNIT      EQU    16D          MAXIMUM NUMBER OF UNITS
  100     RQFL       EQU    100          INITILIZATION FIELD LENGTH
 7700     UINT       EQU    RQFL*100-100   INTILIZATION INTERFACE AREA


          *          UNIT DESCRIPTOR TABLE POINTERS.

                     LOC    0
    1     UXRQ       BSS    1            PPU EXECUTIVE REQUEST
    1     UCIA       BSS    1            CIO REQUEST
    1     UCIB       BSS    1            CIO REQUEST
    1     UCIC       BSS    1            CIO REQUEST
    1     UST1       BSS    1            STATUS (1)
    1     UST2       BSS    1            STATUS (2)
    1     UST3       BSS    1            STATUS (3)
    1     UST4       BSS    1            STATUS (CPU)
    1     UST5       BSS    1            MTS DETAILED STATUS
    1     UST6       BSS    1            MTS DETAILED STATUS AND FORMAT
    1     ULRQ       BSS    1            LAST REQUEST POINTER
    1     UREQ       BSS    1            REQUEST POINTER WORD (INTERNAL)
    1     UFLA       BSS    1            FLAG WORD
    1     UJSQ       BSS    1            JOB SEQUENCE NUMBER, CP NUMBER, VSN INFO
    1     UJBN       BSS    1            UNIT JOB ASSIGNMENT INFORMATION
    1     UUFN       BSS    1            USER NUMBER, FAMILY NAME. (TAPE OWNERSHIP)
    1     UVSN       BSS    1            VOLUME SERIAL NUMBER
    1     UFID       BSS    1            FILE IDENTICATION
    1     UFSN       BSS    1            FILE SECTION NUMBER
    1     USID       BSS    1            SET IDENTIFIER
    1     UGNU       BSS    1            GENERATION INFORMATION
    1     UDAT       BSS    1            DATE INFORMATION
    1     UNITL      BSS    1            LENGTH OF INDIVIDUAL UDT
                     LOC    *O

    7     UUDTL      EQU    UDAT+1-UUFN  LEGNT OF UDT CONTAINING LABEL INFORMATION
```

Figure 9-3. Unit Descriptor Table (UDT) Entry (Level 5) (Continued)

```
***       UNIT DESCRIPTOR TABLE DOCUMENTATION.
*
*T   UXRQ  12/ RS,12/ FUNCTION,12/ MODE,12/ PA,12/ PB
*T,  UCIA  12/ ICID,24/ SKIP COUNT,24/ FET ADDRESS
*T,  UCIB  1/R,1/D,1/,9/ECIO,12/ EOPS,4/LNUM,2/,30/ RECORD REQUEST RET.
*T,  UCIC  12/ FL,24/ FIRST,24/ LIMIT
*T,  UST1  12/ ED,12/ MP,12/ EC,12/ ES,12/ DS
*T,  UST2  24/ DISK PRUS,24/ BLOCK COUNT,12/ USER OPS
*T,  UST3  24/ LAST GOOD RECORD,24/ ERROR PARAMETER,6/ DEN,6/ CV
*T,  UST4  12/ WC,6/OV,6/ UBC,12/ FORMAT,6/ EST,6/ NB,12/ SP
*T,  UST5  60/ MTS DETAILED STATUS
*T,  UST6  36/ MTS DETAILED STATUS,24/MTS FORMAT
*T,  ULRQ  60/ MAGNET LAST REQUEST
*T,  UREQ  12/ 2ONN,12/ ADD,6/ B2,12/ B3,18/ X5
*T,  UFLA  60/ MAGNET FLAGS
*T,  UJSQ   24/ JOB SEQUENCE NUMBER,12/CP NUMBER,6/,18/VSN RANDOM INDEX
*T,  UJBN  42/ JOBNAME,6/ OT,12/
*T,  UUFN  42/ USER NUMBER,6/ FAM,6/ ESW,6/ ACCV
*T,  UVSN  36/ VOLUME SERIAL NUMBER,12/ FLAGS,12/
*T,  UFID  60/ FILE IDENTIFIER
*T,  UFSN  42/ FILE IDENTIFIER CONT.,18/ FILE SECTION NUM
*T,  USID  36/ SET IDENTIFIER,6/ ACCH,18/ FILE SEQUENCE NUM.
*T,  VGNU  30/,12/ GEN VER,18/ GENERATION NUM
*T,  UDAT  30/ CREATION DATE,30/ EXPIRATION DATE.
*
*          DIRECT CELL ALLOCATION RELATIVE TO UDT.
*
*T   UXRQ  12/ RS,12/ FN,12/ MD,12/ PA,12/ PB
*T,  UST1  12/ ED,12/ MP,12/ EC,12/ ES,12/ DS
*T,  UST2  24/ DP - DP+1,24/ BL - BL+1,12/ UP
*T,  UST3  24/ LG - LG+1,24/ EP - EP+1,12/ DC
*T,  UST4  12/ WC,6/ OV,6/ UB,12/ FM,6/ EO,6/ NB,12/ SP
*
*          RS      SEE COMPLETION CODES.
*
*          FN      SEE FUNCTION NUMBERS.
*
*          MD      MODES.
*          0       NONE.
*          1       READ SKIP.
*          2,3     0 - PRU OPERATION.
*                  1 - EOR OPERATION.
*                  2 - EOF OPERATION.
*          3       EOI OPERATION.
*          4       260/264 CONTROL WORD.
*          5       200/204 CONTROL WORD.
```

Figure 9-3. Unit Descriptor Table (UDT) Entry (Level 5) (Continued)

```
*         6      CODED.                                                  COMSMTX
*         10     EOR/EOF THIS OPERATION.                                 COMSMTX
*         12     SET IN = OUT = FIRST.                                   COMSMTX
*         12     REVERSE (READ LABELS ONLY)                              COMSMTX
*         13     REVERSE (READ DATA)                                     COMSMTX
*                                                                        COMSMTX
*         PA,PB  SEE INDIVDUAL FUNCTIONS.                                COMSMTX
*                                                                        COMSMTX
*         UCIA- UCIC          INFORMATION PASSED BY CIO.                 COMSMTX
*                                                                        COMSMTX
*         ICIO   INTERNAL CIO CODE.                                      COMSMTX
*         R      SET IF AUTO RECALL.                                     COMSMTX
*         D      SET IF DATA IN BUFFER.                                  COMSMTX
*         ECIO   USER CIO REQUEST CODE.                                  COMSMTX
*         FOPS   USER,S FET OPTIONS.                                     COMSMTX
*         LNUM   LEVEL NUMBER.                                           COMSMTX
*         FL     JOB FIELD LENGTH.                                       COMSMTX
*                                                                        COMSMTX
*         EQ     EQUIPMENT CONNECT CODE (BITS 13 - 11, 3 -0) AND         COMSMTX
*                CHANNEL DESIGNATOR (BITS 10 - 4)                        COMSMTX
*                                                                        COMSMTX
*         HP     HARDWARE OPTIONS.                                       COMSMTX
*         0      9 TRACK UNIT.                                           COMSMTX
*         1      STATUS 2 AVAILABLE.                                     COMSMTX
*         2      CONVERSION MODE.                                        COMSMTX
*         3      CONTROLLED BACKSPACE.                                   COMSMTX
*         4      PROGRAMMABLE CLIP.                                      COMSMTX
*         5      MTS CONTROLLER.                                         COMSMTX
*         11     BLANK TAPE.                                             COMSMTX
*         12     LAST BLOCK EOR/EOF.                                     COMSMTX
*         13     LAST OPERATION WRITE.                                   COMSMTX
*                                                                        COMSMTX
*         EC     SEE ERROR CODES.                                        COMSMTX
*                                                                        COMSMTX
*         ES     HARDWARE STATUS 2.  (EXTENDED STATUS.)                  COMSMTX
*                ORIGINAL DEVICE STATUS FOR MTS.                         COMSMTX
*                                                                        COMSMTX
*         DS     DEVICE STATUS.                                          COMSMTX
*                FOR MTS, DEVICE STATUS CONVERTED TO 3000 FORMAT.        COMSMTX
*                                                                        COMSMTX
*         UP     USER OPTIONS.                                           COMSMTX
*         0      CODED.                                                  COMSMTX
*         12     NON STANDARD LABELS.                                    COMSMTX
*         13     LABELED                                                 COMSMTX
*                                                                        COMSMTX
*         EP     ERROR PARAMTERS (SEE ERROR PROCESSORS FOR USES.)        COMSMTX
*                                                                        COMSMTX
*         DC     DENSITY AND CONVERSION MODE.                            COMSMTX
*         13 - 6 DENSITY.                                                COMSMTX
*         5 - 0  CONVERSION MODE.                                        COMSMTX
*                                                                        COMSMTX
*         LG     LAST GOOD BLOCK CHECKSUM PREVIOUS BLOCK.                COMSMTX
*         LG+1   LAST GOOD BLOCK CHECKSUM CURRENT BLOCK.                 COMSMTX
*                SEE ROUTINE CKS IN WRITE FOR CHECKSUM METHOD.           COMSMTX
*                                                                        COMSMTX
*         WC     BLOCK WORD COUNT. (0 .LE. WC .LE. 10002)                COMSMTX
*                                                                        COMSMTX
```

Figure 9-3. Unit Descriptor Table (UDT) Entry (Level 5) (Continued)

```
*    OV      OVERFLOW BLACK COUNT. (4000B BYTE BLOCKS)           COMSMTX
*                                                                COMSMTX
*    UB      UNUSED BIT CCUNT.                                   COMSMTX
*                                                                COMSMTX
*    FM      SEE FORMATS.                                        COMSMTX
*                                                                COMSMTX
*    EO      EQUIPMENT WRITTEN ON.                               COMSMTX
*                                                                COMSMTX
*    NB      NOISE BYTE DEFINITION.                              COMSMTX
*    5       FILL OKAY.                                          COMSMTX
*    4 - 0   NUMBER OF BYTES OF NOISE.                           COMSMTX
*                                                                COMSMTX
*                                                                COMSMTX
*    SP      SOFTWARE OPTIONS.                                   COMSMTX
*    0       ABORT RPE/WPE WITH EP SET.                          COMSMTX
*    1       ACCEPT DATA ON RPE/WRE WITHOUT EP SET.              COMSMTX
*    2       INHIBIT ERROR PROCESSING.                           COMSMTX
*    3       RING IN REQUIRED.                                   COMSMTX
*    4       RING OUT REQUIRED.                                  COMSMTX
*    5       INHIBIT UNLOAD.                                     COMSMTX
*    13 - 12  END OF REEL.                                       COMSMTX
*                  0 - READ TO TAPE MARK FOLLOWED BY LABEL       COMSMTX
*                  OR AFTER EOT ON UNLABLEED.                    COMSMTX
*                  1 - ACCEPT BLOCK OF DATA EOF OCCURED ON.      COMSMTX
*                  2      DISCARD BLOCK EOT OCCURED ON.          COMSMTX
*                                                                COMSMTX
*                                                                COMSMTX
*    FLAGS   LABEL PROCESSING FLAGS.                             COMSMTX
*    0       REMOUNT TAPE FLAG.                                  COMSMTX
*    11      FILE OPENED SINCE ASSIGNMENT.                       COMSMTX
*    12      SCRATCH VSN.                                        COMSMTX
*    13      LABEL CHECKING IN PROGRESS.                         COMSMTX
```

Figure 9-3. Unit Descriptor Table (UDT) Entry (Level 5) (Continued)

Note: Words UCIA, UCIB, and UCIC are set up by CIO for MAGNET to process tape I/O requests. The three words are passed to MAGNET by CIO's issuing the TDAM monitor function. The three words are transferred by monitor into the UDT entry specified in the FST entry for the file.

9.1.3 MAGNET - Run-time Executive

Figure 9-4 shows a more detailed outline of the executive code referred to in Figure 9-2. TPRO is a table of processor strings. Each entry is generated at assembly time by the PROC macro which results in a string of processor entry point addresses and/or functions to be processed for a particular request. Part of the TPRO list is indexed by the internal function codes defined in COMSCIO. Thus, any change to COMSCIO may require changes in TPRO. A single entry may be one or two words in length. Each entry may also contain parameters within the string. Up to three 12-bit parameters can be imbedded in a string, but, if less than three are given, the rest are assumed to be zero. A parameter is differentiated from a processor or function by setting bit 11. The three parameters, if specified in a particular string, are referred to as MD, PB, and PA, respectively. These parameters are referenced throughout the listing and in word UXRQ of the UDT entry described in Figure 9-3.

Figure 9-5 shows the relationships between the various subroutines within MAGNET. According to the diagram, MAG calls the major subroutines: CUT, CXR, ASU, and PPU. CUT checks all UDT entries for outstanding requests from CIO. The queue table is also searched for any outstanding requests, and, if any are found, they are processed. CXR is called to check for external requests from DSD. CXR will call MQE to make queue entries for certain requests. ASU is called to perform unit assignment as requested by RESEX. Finally, PPU is called to activate 1MT if a PP is available. Table 9-1 is a list of the functions issued to 1MT by MAGNET.

TABLE 9-1. MAGNET FUNCTIONS TO 1MT

Function Name	Value	Meaning
SED	1	Set Equipment Definition
CUF	2	Complete User FET
MAB	3	Issue message and abort job
FNH	4	Process function (hardware)
PIO	5	Process Interlocked Operation
SKP	6	SKIP
RDF	7	Read data
RLA	10	Read label
WTF	11	Write data
WLA	12	Write label

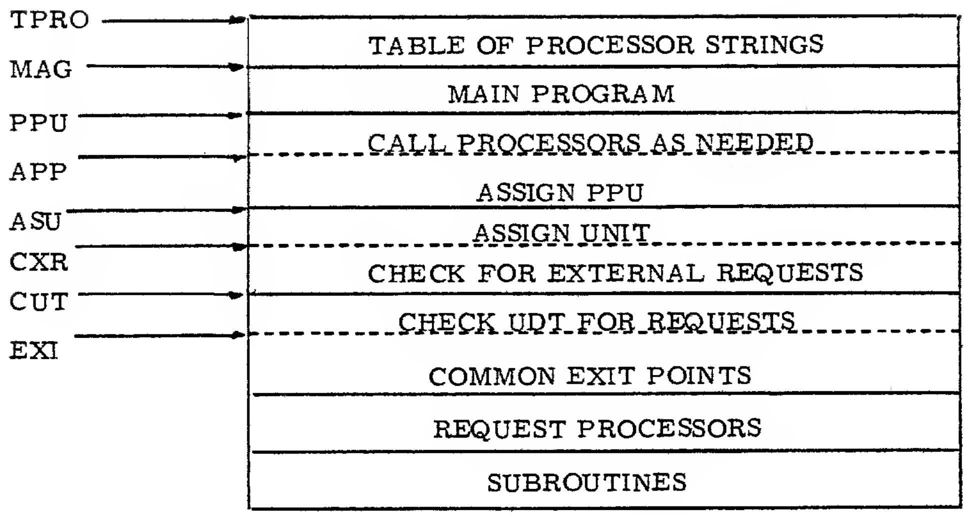


Figure 9-4. Outline of MAGNET Code

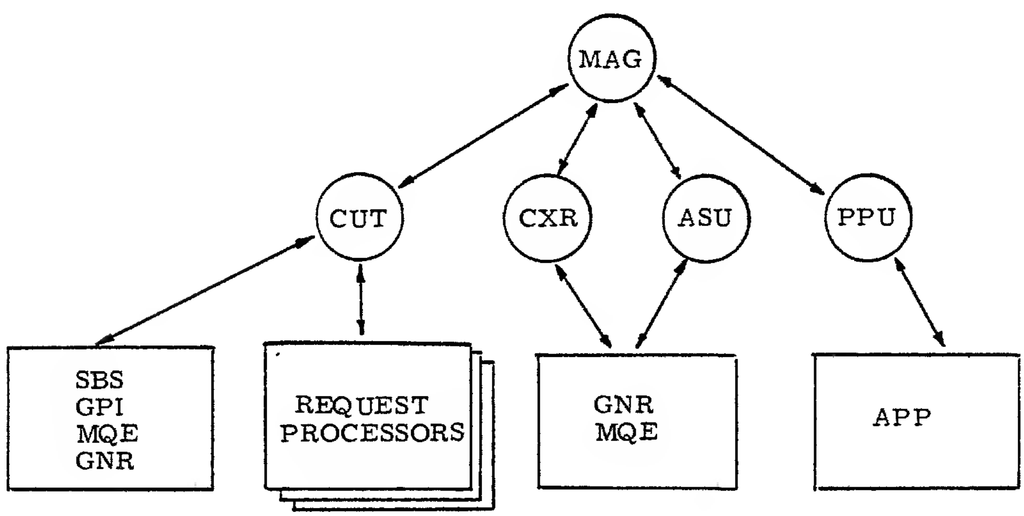


Figure 9-5. MAGNET Subroutine Relationships

These function codes are stored in the first word (UXRQ) of a UDT entry by MAGNET prior to calling 1MT. In the same word, 1MT returns the completion codes in Table 9-2.

TABLE 9-2. 1MT COMPLETION CODES

Code Name	Value	Meaning
RIP	1	Request in Progress
NCP	2	Normal Completion
REQ	3	Re-queue with delay
ERR	4	Error Return

The MAGNET/1MT interlock words are initialized by 1MT and are used by MAGNET to call 1MT. The call to 1MT is via a TLX request in RA+1. The format of an interlock word (UITW) is as follows:

1MT	0	CH	EN	MT
18	6	12	12	12

where:

	bits	meaning
CH	0-4	Channel number
	5	set if 6684 is on channel
	6	set if only one channel access
	7-8	zero
	9-11	processor number (0-7) used by 1MT to index into UITW

EN Equipment number

MT Conversion memory type. The upper bit (11) is set by MAGNET to indicate to 1MT that conversion memory must be loaded.

9.1.4 1MT - PP Magnetic Tape Executive

In general, 1MT searches through the entire UDT to process requests for each unit. As requests are honored, a return code is placed into the first word of each UDT entry. A request is picked up from the first word of the UDT entry and used to scan a table of function code processors. The appropriate overlay is loaded and executed to perform the requested function. Table 9-3 shows which overlay is required to perform the requested function.

TABLE 9-3. FUNCTION OVERLAY

Function	Overlay
SED	3ME
CUF	3ME
MAB	3ME
FNH	3ME
PIO	3ME
SKP	3MF
RDF	3MF
RLA	3MF
WTF	3ML
WLA	3ML

9.2 RESEX - RESOURCE EXECUTIVE

RESEX controls the requests for magnetic tape and removable disk pack resources. The following tape-related control cards are processed:

```
ASSIGN(EQ,LFN,P1,P2,...,PN)
LABEL(LFN,P1,P2,...,PN)
REQUEST(LFN,P1,P2,...,PN)
VSN(LFN=VSN1/VSN2=VSN3)
```

The following control card establishes the maximum number of tape and pack units that will be in use concurrently while the job is running:

```
RESOURC(RT1=N1,RT2=N2,...,RTN=NN)
```

An explanation of the control card parameters is available in Section 5 of the KRONOS 2.1 Reference Manual.

The above control cards (except RESOURC) are available to the user via macro calls. Thus, any job that uses one of the above control cards or macros will initiate a call to RESEX at that job's control point. To avoid destroying the user's field length when RESEX is invoked from a macro call, the special entry point DMP= is defined in RESEX at assembly time. This entry point is used to flag 1AJ to call 1RO prior to loading the RESEX binaries at the control point. 1RO rolls the user's job to a disk file named DM*. This procedure is described in Section 5 of this manual. Other special entry points defined by RESEX are:

ARG=	Suppress 1AJ argument processing
RFL=	Defines RESEX's field length
SSJ=	Declare RESEX to be a special system job

To aid in the allocation of pack and tape resources, RESEX updates two disk files. The two files are known as the resource files and are "fast attach" type direct access permanent files. These files are initialized by ISF under the system user index and assigned the names RESEXDF and RESEXVF. RESEXDF is the resource demand file and contains the maximum concurrent demand for each system resource type. It also contains information for the PREVIEW display and the SHARE table. RESEXVF is the VSN file and contains volume serial numbers associated with a particular job. It also contains a random index to an associated entry in RESEXDF. Entries in the two files are associated with a particular user job and identified by the job's sequence number. Entries in both files are one PRU in length (64D words). Additionally, these two files are updated by the PP routine 0RF. This routine will update a demand file entry or clear entries from either file. It is called at job completion time by 1CJ and by 1TA, REC, and 0DF. 1TA calls 0RF at logout time (for a time-sharing job) to remove a demand file entry for that job.

Entry formats for these two files are defined in common deck, COMSRSX, and are given in Figure 9-6 and 9-7.

Other tables built by RESEX are RET, EVSB, RQ and RDT.

- Resource Equipment Table (RET)

  The resource equipment table consists of a combination of data collected from the EST, MST, and UDT tables. It contains one word entries and is the same length as the EST. The format of an RET entry is as follows:

59 48	47		36	35	24	23	12	11 0
DT	0	CU	0	OU	EQ	NE	EI	flags
12	3	3	3	3	6	6	12	12

where:

DT	Device type from EST entry
CU	Current number of units in chain
OU	Original number of units in chain
EQ	Equipment number (EST ordinal)
NE	Pointer to EST entry of next pack in chain
EI	EVSB index +2 (if any)
flags	as follows:

bit	meaning
0	unit logically assigned
1	not used
2	end of chain of packs
3-10	not used
11	checking labels being done by MAGNET

Label	Entry						Word	Device
RJSQ	Job Sequence Number (24-Bits)						0	
RJBN	Job Name						1	
RVAL	# of Tapes	# of Packs		unused			2	
RMTP	MT	0	Not Used	Unused	Assigned	Demand	3	Seven track
	NT	0					4	Nine track
RRPP	DA1						5	6603
	DB1						6	6638
	DC1						7	863
	DE1						10	ECS
	DF1						11	813/814
	DH1						12	821
	DP1						13	DDP
	DD1						14	854-1
	DD2						15	854-2
	DD3						16	854-3
	DD4						17	854-4
	DI1						20	844-1
	DI2						21	844-2
	DI3						22	844-3
	DI4						23	844-4
	DI5						24	844-5
	DI6						25	844-6
	DI7						26	844-7
	DI8						27	844-8
	MD1						30	841-1
	MD2						31	841-2
	MD3						32	841-3
	MD4						33	841-4
	MD5						34	841-5
	MD6						35	841-6
	MD7						36	841-7
	MD8						37	841-8
RQPD	VSN or Packname	Display Code		MT, DI1, etc.			38	Preview Data
RQPU	User Number			Flags	FST address		39	
RQPT	0			Time			40	
RRPS	Packname			Index	Equip		41	Share Table
	•			•			77	

This table is set up at D/S time only entries needed exist other entries are zero.

Figure 9-6. Demand File Entry (RESEXDF)

where:

RVAL - Contains the validation limits (that is, the number of pack and tape allowed to be assigned to this user). From APUS in control point area.

RMTP - Two words for seven and nine track tape parameters.

RRPP - 27 words of removable pack parameters (left hand 18-bits are in display code.

RRPS - Share table is a list of removable packs assigned to the job.

1 sector per user requesting allocatable devices.

	36	35 24	23 18	17 6	5 0	
VJSQ	24 Bits JOB SEQUENCE	UNUSED				0
VJBN	JOBNAME			UNUSED		1
VLFN	LOGICAL FILE NAME			UNUSED		2
VDFI	UNUSED		DEMAND FILE INDEX			3
VVSN	VOLUME SERIAL NUMBER		UNUSED		CN	4
	• • •			• • •		77

VSNL = 100B

where:

CN - Control byte as follows:

Value	Meaning
/	Multi-reel
=	Alternate reel
0	End of entries

Figure 9-7. VSN File Entry (RESEXVF)

- Environment VSN Buffer (EVSB)

  The environment VSN buffer contains data relating to mounted magnetic tapes and removable packs. An EVSB entry is two words of the following format:

<table>
<tr><th></th><th>48</th><th>47   36</th><th>35   24</th><th>23   18</th><th>17   0</th></tr>
<tr><td colspan="5">VSN</td><td>RI</td></tr>
<tr><td>EQ</td><td>flags</td><td>Sharers</td><td>UDT/MST</td><td colspan="2">Sequence Number</td></tr>
<tr><td>6</td><td>6</td><td>12</td><td>12</td><td colspan="2">24</td></tr>
</table>

where:

RI – Resource index; points to a word in the demand file entry between RMTP and RQPD.

EQ – Equipment number

flags as follows:

bit	meaning
53	assigned
52	scratch VSN
51	counted
48-50	not used

Sharers – Number of users sharing pack

UDT/MST – UDT address if tape or MST address/10B if disk

- Request Block (RQ)

<table>
<tr><th></th><th>36</th><th>35   18</th><th>17   12</th><th>11   6</th><th>5   0</th></tr>
<tr><td>RQ</td><td colspan="2">VSN or PACKNAME</td><td colspan="3">RT</td></tr>
<tr><td>RI</td><td colspan="4">0</td><td>RI</td></tr>
<tr><td>RU</td><td colspan="2">User Job Name</td><td>JO</td><td colspan="2">0</td></tr>
<tr><td>RS</td><td>SN</td><td colspan="4">0</td></tr>
</table>

where:

RT – Resource Type (left justified in the 18-bit field). Values for RT are MT, NT, DA1, DB1, etc. as in the demand file entry on page

RI – Index into demand file entry

JO – Job origin type

SN – Job sequence number

- Resources Demanded Table (RDT)

		42	41				12	11	0
RT			0					F	OE
E1	E2	E3	E4	E5	E6	E7	E8	0	
6	6	6	6	6	6	6	6	12	

where:

RT — Resource Type

F — Flags

OE — Original Equipment

E1-E8 — equipments being demanded from demand file entry in RESB buffer

An outline of the subroutines contained in RESEX follows:

- FET's for:
  - Requested file
  - VSN Entry file (VFILE)
  - Resource Demand file (RESEXDF)
  - VSNFILE (RESEXVF)
  - Two scratch files
- SSJ parameter area
- Control point area parameters
- Temporary storage
- Control card processors:
  - ASSIGN
  - LABEL
  - REQUEST
  - RESOURC
  - VSN
- External request processors:
  - LFM
  - PFM
  - REQ (SCOPE type)
- Resource request Block (RQ)
- Overcommitment Algorithm Control Routine (COMMIT) and subroutines:
  - BRE - Build resource environment
  - BSF - Build scratch file
  - CFU - Check For unit
  - CRC - Check requester complete
  - CRQ - Check request
  - DEI - Demand exceeds installation check
  - OCA - Overcommitment algorithm

- Overcommitment utility subroutines:
  - CAP - Count assigned packs
  - CAT - Count assigned tapes
  - CAU - Count assigned units
  - CDR - Check demand reached
  - DDS - Determine demand satisfaction
  - IAS - Initialize assignments
  - RSB - Read subsystem block
- VSNFILE subroutines (RESEXVF)
  - BVE - Build VSN entry
  - MVE - Make VSNFILE entry
  - SVE - Search for VSNFILE entry
- MAGNETIC tape assignment subroutines:
  - RMT - Request magnetic tape
  - ROA - Request operator assignment
  - VUR - Verify unit request
- Resource demand subroutines:
  - CRV - Check resource validity
  - GRI - Get resource demand entry parameter index
  - RDF - Read demand file
  - UDF - Update demand file
- Preview display subroutines:
  - BPD - Build PREVIEW display
  - EPB - Enter PREVIEW buffer entry
- Utility subroutines:
  - CFA - Check file attached
  - CLB - Clear buffer
  - CUP - Perform timed/event rollout
  - ERR - Error processing
  - GFN - Get family name
  - OPN - Open file
  - CET - Copy EST
  - PER - Process error message
- Common decks
- Buffers (overlay subsequent subroutines)
- Control card pre-processors
  - CCP - Control card pre-processor
  - PCV - Preset control point values

- AMO - Assemble magnetic tape options and call the following processors:

SCD	RTD	SFS	STD
CRD	SCB	SID	STF
FID	SCK	SLT	STK
NMD	SCV	SNS	VSP
RTC	SFA	SPO	WRL

- Control card processing subroutines:
  - AOP - Analyze optional parameters
  - GRD - Generate retention date
  - CJV - Check job validation
  - CLP - Call POP (Pick Out Parameter)
  - FSC - File status check
  - ENF - Enter numeric label field
  - ILF - Initialize label FET
  - SVI - Set VSN index
  - TBD - Build tape block definition
  - VDD - Verify dependent defaults
  - VLC - Validate label call

- External request subroutines:
  - CLF - Convert LFM call to FET
  - CSF - Convert SCOPE call to FET

The two major routines in RESEX are:

- RMT - Request magnetic tape
- COMMIT - Exercise overcommitment algorithm

Entry point processors which call these two major routines are shown in the diagram in Figure 9-8. The actual over commitment algorithm is contained in subroutine OCA The main control routine, however, is entered at COMMIT, as shown in Figure 9-8. RESOURC calls the algorithm routine, OCA, directly with prior calls to BRE and DEI

Subroutine RMT performs tape assignment, calls MAGNET to get a UDT, and builds an FNT/FST entry for the file. The various subroutines called by RMT to perform this function are shown in Figure 9-9.

9.2.1 COMMIT

This is the main program for calling the overcommitment routine, OCA. Prior to calling OCA, the following subroutines are called:

- BRE - Build resource environment (RET)
- CRQ - Builds Request Block (RQ) and builds a demand file entry in buffer, RESB.

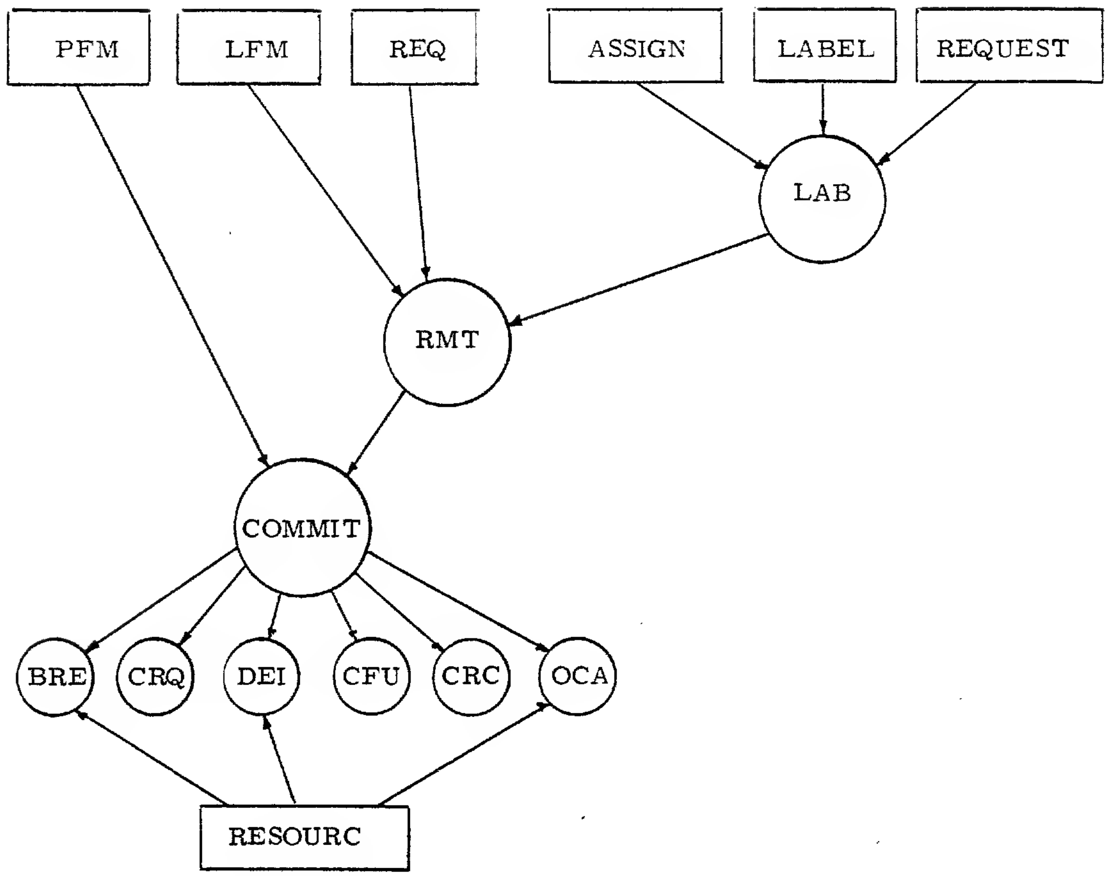


Figure 9-8. Overcommitment Processing

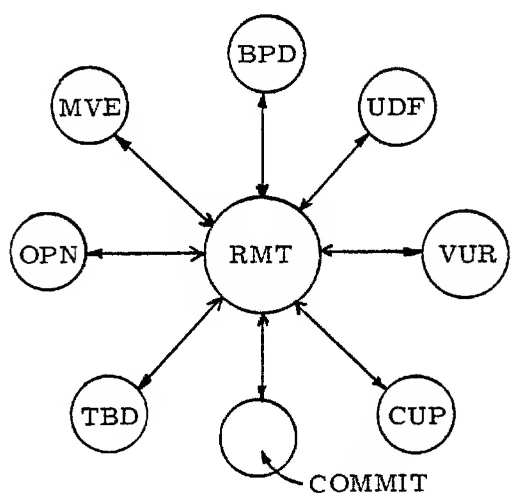


Figure 9-9. Routines called by RMT

- DEI - Searches demand file entry just built in RESB to build the Resources Demanded Table (RDT). Calls DDS to determine if demands are satisfiable. If not, "DEMAND INSTALLATION ERROR".

  DDS determines if resource demands are satisfiable. An attempt is made to assign equipment as follows:

  1. Resource demands are satisfied for larger multi-spindle demands first; tapes last.
  2. A demand is satisfied by a single equipment with a "best fit" determined by the largest spindle residue.
  3. A demand is satisfied by a chain of equipment with no regard to spindle or equipment residue as a "best fit" criterion.
  4. If a demand cannot be satisfied, the previous demand satisfactions are negated and retried (step 2). This is done until all possible chains are exhausted for that resource type.

- CFU - The requested VSN/PACKNAME, if found in EVSB, is assigned to the requestor. If equipment is a removable pack, a SHARE table entry is built. (The entry is part of the demand file entry beginning at RRPS). If the request if for tape, and duplicate VSNs have been declared (VSN, T=A=B=C), they will be used if the original VSN is not found. If more than one VSN is mounted that matches the requested VSN, the operator is given the option to make the assignment.

- CRC - Determines if all demands are satisfied by the resources assigned. If so, the overcommitment algorithm (OCA) is not exercised.

- OCA - Overcommitment Algorithm. Determines if the assignment of the resource to the requestor will cause a potential deadlock. All jobs with assigned resources have their demands written on a scratch file attempting to satisfy outstanding resource demands. If none of the job's demands are satisfiable, then "overcommitment" is said to have occurred. (Subroutine DDS is used to satisfy outstanding demands).

  BSF is called by OCA to build the scratch file containing demand file entries for all jobs with assigned resources.

- 0RF - Updates the demand and assigned counts and SHARE table entries when files on resource devices are returned. When a tape unit is returned, the "assigned" count is decremented by 1. When the last file on a removable pack is returned, the "assigned" count is decremented by 1 and the equipment number in the SHARE table entry is cleared. This causes the entry to be ignored or cleared on subsequent calls. The "demand" count is decremented by 1 only if satisfied.

9.2.2 RMT

RMT is called to request a magnetic tape from MAGNET. The procedure is outlined as follows:

- TBD is called to build Tape Block definition (TB). That is, TBD maps portions of the tape description (FET+10B) into values for use in the RESEX-MAGNET call block. TBD computes the word count, overflow, unused characters, noise size, and fill according to the requested format, frame size, and noise size. TBD also establishes density and conversion mode (BCD, ANSI, or EBCDIC) and validates that the density is proper for the tape type. Finally, TBD ensures that options for ring enforcement or end-of-tape are not conflicting. On exit from TBD, FET+10B is updated to contain density and conversion mode. The format of the Tape Block (TB) definition word is as follows:

59–48	47–36	35–24	23–12		11–0
WC	UC	FO	EO	NO	PO

where:

WC - Word count per tape block
UC - Unused character count and overflow
FO - Tape format:

Value	Format	
0	I	WC = 1000B
1	SI	WC = 1000B
2	X	
3	S	
4	L	
5	E	
6	B	
7	F	

EO - EST ordinal of tape unit
NO - Noise size
PO - Processing options (Refer to Section 7 of the KRONOS 2.1 Reference Manual).

- Store VSN into RQ. (Read from FET+11B.) Then Store jobname and sequence number in RU and RS of request block.

- Call COMMIT to exercise the overcommitment algorithm.

- If the request cannot be satisfied now, RESEX enters timed-event rollout, otherwise VUR is called to validate the unit request and call MAGNET.

- VUR assigns the tape to the requester by sending the "call block" to MAGNET. First, the UDT is read (from MAGNET) to verify that the VSN has not changed, and that the unit has not been assigned to another job. Next, the conversion field in the UDT is checked to ensure that MAGNET could make the requested conversion change. If the tape is being assigned automatically, LFM is called to assign the equipment. Next,a callblock is built from tape descriptors (FET+10B), VSN information, and other UDT information. If the tape is a KRONOS 2.1 labeled tape, the accessibility is verfied (see description of file accessibility on page 7-41 of the KRONOS Reference Manual). The format of the block is as follows:

59–36	35–24	23–12	11–0
Interlock			
		UO	UDT address
			D | C
TB			
SN		VI	VA
Jobname			OT |
VSN			

where:

- UO - User options
- D - Density
- C - Conversion type
- TB - Tape block definitions established by subroutine TBD
- SN - Job sequence number
- VI - VSN index
- VA - VSN random address
- OT - Job origin type

The call block is sent to MAGNET. After Magnet responds, the UDT is reread to verify that the tape has been assigned correctly. If not, the call is repeated. If MAGNET has assigned the tape, VUR builds the tape file FNT/FST according to the following format:

	59–48	47–36	35–24	23–12	11–0
FNT	File Name				Type | CP
FST	ID | EQ	UDT adr	F | T	VA	labeled flag |

where:

- F - Format (0-7)
- T - Type (0-7)
- VA - VSN random address

Finally, VUR issues the assignment message:

"xxxx ASSIGNED TO ffffff, VSN = nnnnnn."

where:

- xxxx = Unit number
- f-f = File name
- n-n = Volume serial number

- RMT will then call UDF to update the demand file using the entry in the RESB buffer. UDF also updates word UJSQ of the control point area to contain the demand file random index for this entry.

- RMT then calls BPD to build the PREVIEW display (E.P.). This information is sent to MAGNET's field length where DSD can read it for the display. BPD builds entries for the PREVIEW buffer by using the RQPD field of the demand file entry. The first word in the buffer contains the length of the buffer. Entries are two words and are in a hierarchy based on the number of additional units required and the maximum units returned at job completion time. The first entry is the last requestor (if any). Entries might also contain data from MAGNET UDTs when ring-enforcement requires remounting the tape or when VSNs are needed for multi-reel processing. The entry format is as follows:

RQPD is one of the two following formats depending on whether its a Tape or a PACK.

RQPD word			
USER NUMBER			
42		2	10

FST address

where:

RQPD is the RQPD word from the demand file entry with one of the following formats:

	59 … 18		17 … 0
PACKS	Pack Name		RT
TAPES	VSN	R	RT

where:

R - Ring-in flag
RT - Resource type (MT, NT, DB1, etc.)

The following is a dump of MAGNETs FL and the RESEXDF and RESEXVF files.

TABLE 9-4. MAGNET - RESEX

Foil	Word	Description
1		Picture of E,T and E,P display and job sequence.
2	127703	Start of MAGNET fl. Refer to figure 9-2 WORKSHOP manual 6 UDTs, start at 1060 and end at 1264.
	127704	Queue start at 1265
	127706	XREQ empty
	127710	1MT interlock words
	127722	FL status word
	127723	TFUN 1MT function table
	127735	RCAL see p. 9-21 Workshop manual interlock = 0
	127736	UDT address = 1106,UO=4
	127737	D/C = 300
	127740	TB = 100510400
	127741	SN = AABF, VA = 1
	127742	Jobname = JOB1 ABFA
	127743	VSN = ONE001
2	127744 & 127745	Unused
	127746	VSN=SIX, eq=DI1 UN=MLO, FST addr 3313
	127747	Start of E,P display
	127750	VSN=FIVE, EQ = MT
	127751	UN=MLO, FST adr=3331
	127752	USN=TWO001, EQ=MT
	127753	UN=MLO, FSTadr =3323

TABLE 9-4. MAGNET - RESEX (Continued)

Foil	Word	Description
3	130760 = (relative 1060)	Start of UDT See 9-6 Workshop
	130764	See COMSMTX UST1
	130767	See COMSMTX UST4
	130772	See COMSMTX ULRQ
	130775	Job Seq = AABG CP=3
	130776	Jobname = JOB2ABGA
	131000	VSN = ****50
	131001	FI=unlabeled
	131004	Generation number = 1
	131005	UDAT = date
	131165	Queue table
3	131171	end of Queues (not shown)
	131177	end of MAGNET (not shown)

TABLE 9-4. MAGNET - RESEX (Continued)

Foil	SE	Description
4	0	Start of RESEXDF see figure 9-6, Workshop Manual
4	1	RJSQ = AABC
		RJBN = JOB5ABCA
		RVAL = MT = 4, NT = 4, from VALIDUX
		RMTP = DI1 assign = 0, req = 1
		RQPD = SIX, type = DI1
		RQPU = MLO fst addr = 3313
		RQPT = 40723134220

TABLE 9-4. MAGNET - RESEX (Continued)

Foil	SE	Description
4	2	RJSQ = AABF RJBN = JOB1ABFA RVAL = MT = 4, NT = 4 from VALIDUX RMTP = MT assign = 1, req = 2 from RESOURC card RQPD = TWO001 type = MT RQPU = MLO Fst addr = 3323 RQPT = time = 40723134710
5	3	RJSQ = AABH RJBN = JOB3ABHA RVAL = MT = 4, NT = 4 RMTP assign = 0, req = 1, no RESOURC card RQPD = FIVE type = MT, etc.
5	4	JOB2ABGA RMTP MT assign = 1, req = 1, etc, No RESOURCE card
5	4	EOR Job5 was an old entry and is ignored since this EOR is an empty sector.
	40	EOI
6		Start of RESEXVF, see figure 9-13.
6	1	VJSQ = AABF VJBN = JOB1ABFA VLFN = ONE VDFI = 1 VVSN = ONE001
6	2	VJBN = JOB3ABHA VLFN = FIVE VDFI = 2

Note: 1 sector/job using allocatable devices.

TABLE 9-4. MAGNET - RESEX (Continued)

Foil	SE	Description
7	3	VJBN = JOB2ABGA VLFN = THREE VDFI = 3
7	4	VJBN = JOB1ABFA VLFN = TWO VDFI = 1 Note: same job as one above.
7	5	EOR Job 4 is an old entry and is ignored.
7	40	EOI - There is no entry for SIX for JOB5 since it is not yet assigned.

```
JOB1,CM30000,T7777.
ACCOUNT,MLO,MLO.
RESOURC(MT=2)
REQUEST(ONE,VSN=ONE001)
REQUEST(TWO,VSN=TWO001)
0000000000000000000000
JOB2,CM30000,T7777.
ACCOUNT,MLO,MLO.
REQUEST(THREE)
REQUEST(FOUR)
0000000000000000000000
JOB3,CM30000,T7777.
ACCOUNT,MLO,MLO.
LABEL(FIVE,FI=FIVEONE,SI=FIVETWO,VSN=FIVE)
0000000000000000000000
JOB5,CM30000,T7777.
ACCOUNT,MLO,MLO.
ATTACH(SIXEX/PN=SIX,R=DI1,NA)
0000000000000000000000
```

1

E,T

EQ	VSN	DEN	R	F	CP	JOB	STATUS
MT50	****50	800		X	4	JOB2ABGB	LOADPT
	UNLABELED				RN	1.	
MT51	ONE001	800		X	,	JOB1ABFB	ROLLED
	UNLABELED				RN	1.	
MT52		800					IDLE
MT53		800					IDLE
NT60	****60	1600					IDLE
	UNLABELED				RN	1.	
NT61		1600					IDLE

E,P

NO (FNT ord)	EQ	PN/VSN	UN	RING
14	MT	FIVE	MLO	
11	MT	TWO001	MLO	
5	DI1	SIX	MLO	

MAGNET FL

(2)

```
ABSOLUTE DUMP FROM 127700        TO        131000        PAGE        1

127700      00000000000000000000                    00000000000000000000                    00000000000000000000
127703      00000000125400001060   F  J*  M≡        00000000000000001265          J*        00000000000000000000
127706XREQ  00000000000000000000                    00000000000000000000                    34152400111300050002   1H7 1K E B
127711      00002565000005251060   U*  EUH≡         40000000000000000000   5                40000000000000000000   5
127714      40000000000000000000   5                40000000000000000000   5                40000000000000000000   5
127717      40000000000000000000   5                40000000000000000000   5                40000000000000000000   5
127722      00000013000000000001   K     A          22003124000000000000   R YT             22202144000000000000   RP09
127725      22202233000000000000   RPR0             22002374000000000000   H SS             24452403000000000000   T+TC
127730      34202144000000000000   1PU9             24602144000000000000   T≡U9             24432650000000000000   TBV/
127733      44602144000000000000   9≡Q9             44433056000000000000   9AX,             00000000000000000000
127736      00000000040000001106       5  IF        00000000000000000300          C         10000000000051040000   H     (D
127741      01010206000000000001   AABF     A       12170234010200010000   JOB1ABFA         17160533333400000000   ONE001
127744      00000000000000000000                    00000000000000000000                    23113000000000041134   SIX    011
127747      15141700000000003313   MLO     OK       06112605555500152400   FIVE   M7        15141700000000003331   MLO     nY
127752      24271733334400152400   TWO001 MT        15141700000000003323   HLO     0S       00000000000000000000
127755      00000000000000000000                    00000000000000000000                    00000000000000000000
127760      00000000000000000000                    00000000000000000000                    00000000000000000000
127763      00000000000000000000                    00000000000000000000                    00000000000000000000
127766      00000000000000000000                    00000000000000000000                    00000000000000000000
127771      00000000000000000000                    00000000000000000000                    00000000000000000000
127774      00000000000000000000                    00000000000000000000                    00000000000000000000
127777      00000000000000000000                    00000000000000000000                    00000000000000000000
130002      00000000000000000000                    00000000000000000000                    00000000000000000000
130005      00000000000000000000                    00000000000000000000                    00000000000000000000
130010      00000000000000000000                    00000000000000000000                    00000000000000000000
130013      00000000000000000000                    00000000000000000000                    00000000000000000000
130016      00000000000000000000                    00000000000000000000                    00000000000000000000
130021      00000000000000000000                    00000000000000000000                    00000000000000000000
130024      00000000000000000000                    00000000000000000000                    00000000000000000000
130027      00000000000000000000                    00000000000000000000                    00000000000000000000
130032      00000000000000000000                    00000000000000000000                    00000000000000000000
130035      00000000000000000000                    00000000000000000000                    00000000000000000000
130040      00000000000000000000                    00000000000000000000                    00000000000000000000
130043      00000000000000000000                    06370705622407060000   F4GE1TGF         02210000000000000000   BQ
130046      02210000000000000000   BQ               02210000000000000000   BQ               06200567066000040000   FPEAF≡ n
130051      06500155000000000000   F/A              05420650000000000000   E/F/             06360654620200000000   F3F*1B
130054      01650000000000000000   A*               00064000400040006160    F5 5 5 1≡       00104100440040000652    H6 9 5 F)
130057      06200000000000000000   FP               06520620000000000000   F)FP             05310003053300000000   EY CEO
130062      06000000000000000000   F                01610604000000000000   A(F0             06330000000000000000   F0
130065      05640000000000000000   E≠               00010214000000000000    ABL             00044000400300000000    0S SC
130070      01610550017200000000   A(E/A<           01720000000000000000   A<               02160004400040020560   BN 0S 5AF≡
130073      00104100450040006176    H6 + 5 (¬       00040562001041004501    0E) H6 +A       40006176000400000000   5 1- 0
130076      02230170000000000000   BSA+             06160010610043004001   FN H( B 5A       00104100430040010000    H6 B 5A
130101      06130616053500000000   FKFNEZ           00104100410040006201    H6 6 5 1A       02100153000000000000   BHA$
130104      07030233020500000000   GCBUBE           00034000400200020672    C5 5B BF<       01700000000000000000   A+
130107      02330670017000000000   BOF+A+           05350572070302330217   EZE<0CH0R0       00000000000000000000
130112      00104100410040006201    H6 6 5 1A       02100221000000000000   BMBU             00044000400200104100    0S 5B H6
130115      00040526000000000000    UEV             05670004000000000000   EA 0             00034000400206720170    C5 5BF<A+
130120      00000000000000000000                    05670636070062120616   EAF30 1JFN       00000000000000000000
130123      05670004400040010000   EA 0S 5A         05720703023300124100   E<0CB0 J6        42004000021701440000   7 5 B0A9
130126      00124100410000106100    J6 6  H(        43004004000000000000   B 50             00104100420000000000    H6 7
130131      07030233001241004200   0CBU J6 7        02050000000000000000   BE               00044000400400000000    0S 50
130134      02260002000000000000   BV B             00044000400501700000    0S 5EA+         00124100430000000000    J6 B
130137      01000010516100046000   A  H(( 0≡        71602411115711000270   +≡T1M+M B+       20652126617110200005   P*0V1+HP E
130142      20130126610100001054   PKAV1A  H*       51300027051100000265   (X B+(H R*       13777213445170000045   K(<K9(+  +
130145      55211267713763721601    QJA++1<NA       03060002536157777577   CF B$[.1>1       66671055000025066610   VAM   UFVH
130150      76460665106676027763   ¬-F*HV¬B((       20473665550324000251   P*3* CT B(       77651507100000020601   1*MOH  BFA
130153      54620206660326000255   =1BFVCV B        01000010136100046000   A  HK[ 0≡        01000042161000046000   A  001 0≡
```

UITW

TFUN

RCAL

PBUF

Code

MAGNET FL

(3)

```
ABSOLUTE DUMP FROM 127700     TO     131000             PAGE      4
130712  54661020000101546000  =VHP AA=≡   04000002550000000000  D  B         51100012651061154610  (H J*H[H-H
130715  54111106110311001012  =1IF1C1 HJ  03210010155466154661  CQ HH=vH-[   70460001055130000022  += AE(X  R
130720  21406204062133637634  Q51DFQ0;¬1  03060010137160220610  CF HK+≡RFL   20652712000002220222  P*WJ  BRBR
130723  36662731401266143101  3vWYSJvLYA  20151366610100001054  PMKv1A  H=   02000010136100046000  B  HK1 DE
130726  20560124543661454610  P,A7=31+-H  00000000000000000000               50100000035020000002  /H  C/P  B
130731  73510211307341137654  >(B1X>6K¬=  71100010000570001034  +H H E+ H1   37461033400103410611  4=H05AC6FI
130734  73520031500103610511  >1 Y/AC1E1  04700010367256000000  D+ H3<,      37465032400104120201  4=/Z5ADJBA
130737  21263724277777346000  QV4TW##1≡   03040010417256000000  CD H6<,      72657767765010000007  <*IA¬/H  G
130742  43230714000000015112  8SGL   A(J  03360010267160000634  C3 HV+≡ F1   73260734502750524505  >VQ1/W/1+E
130745  27606246064465626676  W≡1-+9*1v¬  22676422623754273460  RA#R14=W1≡   20452020000102646000  P+PP AB*≡
130750  20652010000105446000  P*PH AE9≡   04000002400000000000  D  B5        71602203140400001050  +≡RCLD  H/
130753  01300000006100046000  AX   [ D≡   04000002430000000000  D  BB        56610020000105346000  ,1 P AE1≡
130756  00000000210000010300     W  AC    07600041000000000000  G≡ 6         00000000000000000000
130761  00000000000000000000              00000000000000000000               00000000000000000000
130764  60200030003000001221  ≡P X X  JQ  00000000000000000000               00000000000000000300           C
130767  10000000000250044000  H    B/D5   00000000000000000000               00000000000000000000
130772  00020004000000000000   B D        00000000000000000000               00000004000000000002      D    B
130775  01010207000400000000  AABG D      12170235010207010000  JOB2AB0A     55555555555555000055
131000  47474747403300060000  ****50 F    25161401020514050457  UNLABELED.   55555555555555000001           A
131003  55555555555555000001           A  00000000000000000001           A   42373533334237353333  7420074200
131006  00000000000000000000              00000000000000000000               00000000000000000000
131011  00000000000000000000              60210030001700001221  ≡Q X O  JQ   00000000000000004000           S
131014  00000000000000000300           C  10000000000051040000  H     (D     00000000000000000000
131017  00000000000000000000              00020004000000000000   B D         00000000000000000000
131022  00000000000000000000              01010206000000000001  AABF     A   12170234010206010000  JOB1ABFA
131025  55555555555555000055              17160533333400020000  ONE001 B     25161401020514050457  UNLABELED.
131030  55555555555555000001           A  55555555555555000001           A   00000000000000000001           A
131033  42373533334237353333  7420074200  00000000000000000000               00000000000000000000
131036  00000000000000000000              00000000000000000000               64200000000000000000  #P
131041  00000000000000000000              00000000000000000000               00000000000052000000        )
131044  00000000000000000000              00000000000000000000               00000000000000000000
131047  00000000000000000000              00000000000000000000               00000000000000000000
131052  00000000000000000000              00000000000000000000               00000000000000000000
131055  00000000000000000000              00000000000000000000               00000000000000000000
131060  00000000000000000000              00000000000000000000               00000000000000000000
131063  00000000000000000000              00000000000000000000               00000000000000000000
131066  64210000000000000000  #Q          00000000000000000000               00000000000000000000
131071  00000000000053000000        S     00000000000000000000               00000000000000000000
131074  00000000000000000000              00000000000000000000               00000000000000000000
131077  00000000000000000000              00000000000000000000               00000000000000000000
131102  00000000000000000000              00000000000000000000               00000000000000000000
131105  00000000000000000000              00000000000000000000               00000000000000000000
131110  00000000000000000000              00000000000000000000               00000000000000000000
131113  00000000000000000000              50200037001710001321  /P 4 OH KQ   00000000000000000000
131116  00000000000000000000              00000000000760000000      G≡       00000000000000000000
131121  00000000000000000000              00020004000000000000   B D         00000000000000000000
131124  00000000000000000000              00000000000000000000               00000000000000000000
131127  55555555555555000055              47474747413300060000  ****60 F     25161401020514050457  UNLABELED.
131132  55555555555555000001           A  55555555555555000001           A   00000000000000000001           A
131135  42373533334237353333  7420074200  00000000000000000000               00000000000000000000
131140  00000000000000000000              00000000000000000000               50210037000000001300  /Q 4     K
131143  00000000000000004000           S  00000000000000000302          CB   00000000000761000000        G(
131146  00000000000000000000              00000000000000000000               00020001000000000000   B A
131151  00000000000000000000              00000000000000000000               00000000000000000000
131154  00000000000000000000              00000000000000000000               00000000000000000000
131157  00000000000000000000              00000000000000000000               00000000000000000000
131162  00000000000000000000              00000000000000000000               40000000000000000000  S
131165  00000000000000000000              77777777777777777777  ##########   77777777777777777777  ##########
```

UDT #1

UDT #2

UDT #3

UDT #4

UDT #5

UDT #6

Start of queue Table which is empty

END OF MAGNET FL at 127700 +1300 FL = 131200

9-30

DUMP7K(TK=266)   RESEXDF   DUMP7K - VER. 1   74/07/19. 11.40.04.   PAGE 1   (4)

```
WORD  7K=266  SE=0    B1=3777 B2=77           7K=266  SE=1     B1=2    B2=100              TK=266  SE=2     B1=3    B2=100
 0    22052305300406001300  RESEXDF K         01010203000000000000  AABC                   01010206000000000000  AABF
 1    00004266114417013307    7Y190A0G        12170240010203010000  JOB5ABCA               12170234010206010000  JOB1ABFA
 2    00000000040723132312      DG5K5J        00040004000000000000   D D                   00040004000000000000   D D
 3    71261405600130033201  +VLE≡AXCZA        00000000000000000000                         15240000000000010002  H7      A B
 4    34023040051554007076  1BX≡EM= +¬        00000000000000000000                         00000000000000000000
 5    30431277310160033405  XBJ1YA≡C1E        00000000000000000000                         00000000000000000000
 6    30071003140660400313  XGHCNF≡5CK        00000000000000000000                         00000000000000000000
 7    14045400707637020603  LD= +¬4BFC        00000000000000000000                         00000000000000000000
10    22052305300406377777  RESEXOF4;1        00000000000000000000                         00000000000000000000
11    40070000000242664000  50    87V5        00000000000000000000                         00000000000000000000
12    00000000040621104646      DFQH--        00000000000000000000                         00000000000000000000
13    00000000040621104646      DFQH--        00000000000000000000                         00000000000000000000
14    00000000040621104646      DFQH--        00000000000000000000                         00000000000000000000
16    00000000000000003307          0G        00000000000000000000                         00000000000000000000
20    70760510301404041070  +¬EMXLDDH+        04113400000000000001  DI1        A           00000000000000000000
21    12010503000000013014  JAEC    AXL       00000000000000000000                         00000000000000000000
22    34033007100316046010  1CAGHCND≡H        00000000000000000000                         00000000000000000000
23    37135400714730261074  4K= +*XVHS        00000000000000000000                         00000000000000000000
24    22000017160134043111  R  UNA1UY1        00000000000000000000                         00000000000000000000
25    34063003100607333011  1FXCHF00X1        00000000000000000000                         00000000000000000000
26    34063007100316203272  1FXGHCNPZ<        00000000000000000000                         00000000000000000000
27    54007177106323002100  = +IHIS W         00000000000000000000                         00000000000000000000
30    54007176300610752100  = +¬XFH2U         00000000000000000000                         00000000000000000000
31    66206010300612033414  VP≡HXFJC1L        00000000000000000000                         00000000000000000000
32    50140010340637040563  /L H1F4DE1        00000000000000000000                         00000000000000000000
33    30053201340530070100  XEZA1EXGA         00000000000000000000                         00000000000000000000
34    70030100721754007231  +CA <D= <Y        00000000000000000000                         00000000000000000000
35    14056001300332013402  LE≡HXCZA1B        00000000000000000000                         00000000000000000000
36    20007230310160033405  P <XYA≡C1E        00000000000000000000                         00000000000000000000
37    30071003160661707321  XGHCNF1+>W        00000000000000000000                         00000000000000000000
40    50007324137733105400  / >1K1OH=         23113000000000041134  SIX     DI1            24271713333400152400  TW0001 MT
41    73240306370207443101  >TCF4BG9YA        15141700000000003313  MLO      0K            15141700000000003323  MLO      05
42    60033405300310711251  ≡C1EXCH+J(        00000000040723134220       DG5K/P            00000000040723134710       DG5K*H
43    11500566300610060763  I/EVXFHFG1        00000000000000000000                         00000000000000000000
44    30071003160560103010  XGHCNE≡HXH        00000000000000000000                         00000000000000000000
45    12340554300710031606  J1E*XGHCNF        00000000000000000000                         00000000000000000000
46    60101400340450047321  ≡HL 1D/U>Q        00000000000000000000                         00000000000000000000
47    53040010054136041104  $D HE63D1D        00000000000000000000                         00000000000000000000
50    05703005320134053007  E+X≡ZA1EXG        00000000000000000000                         00000000000000000000
51    01007217000000000000  A <O              00000000000000000000                         00000000000000000000
52    00000000100073262000      A >VP         00000000000000000000                         00000000000000000000
53    11130200131450001100  IKB KL/ 1         00000000000000000000                         00000000000000000000
54    05100200120316020320  EHB JCNBCP        00000000000000000000                         00000000000000000000
55    14020100131002006305  LBA KHB 1E        00000000000000000000                         00000000000000000000
56    04720200635004061400  D<B 1/DFL         00000000000000000000                         00000000000000000000
57    34571444010013103057  1.LYA KHX.        00000000000000000000                         00000000000000000000
60    60201701601056001312  ≡POA≡H, KJ        00000000000000000000                         00000000000000000000
61    30131204055130201277  XKJDE(XPJ1        00000000000000000000                         00000000000000000000
62    34055100055160103010  1E( E(≡HXH        00000000000000000000                         00000000000000000000
63    10060704140601001310  HFGDLFA KH        00000000000000000000                         00000000000000000000
64    02000547301410031620  B E*XLHCNP        00000000000000000000                         00000000000000000000
65    32725400460610632300  Z<= -FH15         00000000000000000000                         00000000000000000000
66    21005400460530220504  Q = -EXRED        00000000000000000000                         00000000000000000000
67    14040100131030213406  LDA KHXQ1F        00000000000000000000                         00000000000000000000
70    02004577300234273003  B +(XB1WXC        00000000000000000000                         00000000000000000000
71    17013430060336303727  O41AFC3X4W        00000000000000000000                         00000000000000000000
```

97404700C

RESEXDF

```
                   DUMP7K(TK=266)                                               DUMP7K - VER. 1     74/07/19. 11.40.04.   PAGE   2        (5)
 72      30213422140134233024  AQIMLA1SKT     00000000000000000000                         00000000000000000000
 73      13771105342401007326  K(1E1TA >V     00000000000000000000                         00000000000000000000
 74      00000000000006366322        F31R     00000000000000000000                         00000000000000000000
 75      03150010640443001063  CM M#DB HI     00000000000000000000                         00000000000000000000
 76      675=5055200106466300  A=/ PAF-(      00000000000000000000                         00000000000000000000
 77      43400043000100410677  #5 o AF6F(     00000000000000000000                         00000000000000000000

WORD     TK=266   SE=3    B1=4   B2=100       TK=266   SE=4    B1=5   B2=100               TK=266   SE=5   B1=6     B2=0       EOR
  0      01010210000000000000  AADM           01010207000000000000  AABG                   01010215000000000000  AABM
  1      12170236010210010000  JOD3ABHA       12170235010207010000  JOB2AB0A               12170240010215010000  JOB5ABMA
  2      00040004000000000000   D D           00040004000000000000   D D                   00040004000000000000   D D
  3      15240000000000010001  MT        A    15240000000000010001  MT      A A            00000000000000000000
 20      00000000000000000000                 00000000000000000000                         04113400000000010001  D11     A A
 40      06112605555500152400  FIVE    M7     00000000000000000000                         00000000000000000000
 41      15141700000000003331  MLO       0Y   15141700000000003325  MLO      0U            15141700000000003315  MLO      0M
 42      00000000040723134307      DGSK8G     00000000040723134657      DGSK-.             00000000040723134567      DGSK*A
 43      00000000000000000000                 00000000000000000000                         02111414000000200001  BILL    P A

WORD     TK=266   SE=36   B1=37  B2=100       TK=266   SE=37   B1=40  B2=0                 TK=266   SE=40  B1=0     B2=0       EOI
  0      00000000000000000000                 00000000000000000000                         00000000257600003242      U¬  Z7
  1      00000000000000000000                 00000000000000000000    EOR                  01532205230530040600  ASHESEXDF
  2      00000000000000000000                 00000000000000000000                         00240002766500000311   T R¬*  CI
  3      00000000000000000000                 00000000000000000000                         17430010000001011245  OB H  AAJ*
  4      00000000000000000000                 00000000000000000000                         00323413000020010000   ZOK  PA
  5      00000000000000000000                 00000000000000000000                         25770000237600003002  U¬  S¬  XB
  6      00000000000000000000                 00000000000000000000                         00010100100000030600   AA H  CF
  7      00000000000000000000                 00000000000000000000                         30503051305214771701  X/X(X)L(QA
 10      00000000000000000000                 00000000000000000000                         05763075605030511237  F¬X2≡/X(J4
 11      00000000000000000000                 00000000000000000000                         10073474305113771006  HG1$X(K(HF
 12      00000000000000000000                 00000000000000000000                         33500462100602000135  O/D)HFB A2
 13      00000000000000000000                 00000000000000000000                         02000424011500055400  B DTAM E=
 14      00000000000000000000                 00000000000000000000                         01333001101431026114  AOXAHLYB(L
 15      00000000000000000000                 00000000000000000000                         15420100053610145400  H7A E3HL=
 16      00000000000000000000                 00000000000000000000                         03261063230023005400  CVH(S S H
 17      00000000000000000000                 00000000000000000000                         03252001301402000335  CUPAXLB C2
 20      00000000000000000000                 00000000000000000000                         05522002140102000335  E)PBLAB C2
 21      00000000000000000000                 00000000000000000000                         05041464617077745400  FDL≠[+(S=
 22      00000000000000000000                 00000000000000000000                         02721702341750000533  B<DB1D/ E0
 23      00000000000000000000                 00000000000000000000                         54000275500001355400  = B2/ A2=
 24      00000000000000000000                 00000000000000000000                         03015000032554000305  CA/ CU= CE
 25      00000000000000000000                 00000000000000000000                         50000326540003061422  / CV= CFLR
 26      00000000000000000000                 00000000000000000000                         02000364307660030200  B C≠X¬≡CB
 27      00000000000000000000                 00000000000000000000                         05475600057603132000  E*, E¬CKP
 30      00000000000000000000                 00000000000000000000                         04773517300710060607  D(2OXGHFFG
 31      00000000000000000000                 00000000000000000000                         10713406140034070200  H+1FL 1GB
 32      00000000000000000000                 00000000000000000000                         06064017341650170001  FF5D1N/U A
 33      00000000000000000000                 00000000000000000000                         34033017020006161057  1CXOB FNH.
 34      00000000000000000000                 00000000000000000000                         34014017340730164417  1A5O1GXN90
 35      00000000000000000000                 00000000000000000000                         36174017107110213303  3O5OH+HU0C
 36      00000000000000000000                 00000000000000000000                         44170605300105030100  9OFEXAECA
 37      00000000000000000000                 00000000000000000000                         02213004020004462000  BQXDB U¬P
 40      00000000000000000000                 00000000000000000000                         13653415200013015400  K+1MP KA=
 41      00000000000000000000                 00000000000000000000                         05332000053654000135  EOP EJ= A2
 42      00000000000000000000                 00000000000000000000                         14002310362001010134  L SH3PAAA1
 43      00000000000000000000                 00000000000000000000                         30143502106335011014  XL2BH(2AHL
 44      00000000000000000000                 00000000000000000000                         31026010301113771006  YB≡HX1K(HF
 45      00000000000000000000                 00000000000000000000                         33100410230435030561  OHDM5O2CE[
 46      00000000000000000000                 00000000000000000000                         30120502301534150100  XJEBXM1MA
```

9-32

DUMPTK(TK=267)   RESEXVF   DUMPTK - VER. 1   74/07/19. 11.40.15.   PAGE 1   6

WORD	TK=267 SE=0 B1=3777 B2=77		TK=267 SE=1 B1=2	B2=100	TK=267 SE=2 B1=3	B2=100
0	22052305302606001300	HESEXVF K	01010206000000000000	AABF	01010210000000000000	AABH
1	00004267114417013311	7^190A01	12170234010206010000	JOB1ABFA	12170236010210010000	JOB3ABHA
2	00000000040743132313	0G5K5K	17160500000000000000	ONE	06112605000000000000	FIVE
3	71261405600130033201	+VLE≡AXCZA	00000000000000000002	B	00000000000000000003	C
4	34023040051554007076	1BX>EH= ↑¬	17160533333400000000	ONE001	06112605555500000000	FIVE
5	30431277310160033405	XBJ+YA≡C1E	00000000000000000000		00000000000000000000	
6	30071003160660400313	XGHLNF≡5CK	00000000000000000000		00000000000000000000	
7	14045400707637020603	LO= ↑¬4BFC	00000000000000000000		00000000000000000000	
10	22052305302606377777	HESEXVF4↑↑	00000000000000000000		00000000000000000000	
11	40070000000242674411	5G B7∧91	00000000000000000000		00000000000000000000	
12	00000000040621104647	DFQH-*	00000000000000000000		00000000000000000000	
13	00000000040621104647	OFQH-*	00000000000000000000		00000000000000000000	
14	00000000040621104647	OFQH-*	00000000000000000000		00000000000000000000	
16	00000000000000003311	01	00000000000000000000		00000000000000000000	
20	70760510301404041070	↑¬EΠXLDOH↑	00000000000000000000		00000000000000000000	
21	12010503000000013014	JAEÇ AXL	00000000000000000000		00000000000000000000	
22	34033007100316046010	1CXŲHCND≡H	00000000000000000000		00000000000000000000	
23	37135400714730261074	4K= +*XVHS	00000000000000000000		00000000000000000000	
24	22000017160134043111	H UHA1DY1	00000000000000000000		00000000000000000000	
25	34063003100607333011	1FXLHFG0X1	00000000000000000000		00000000000000000000	
26	34063007100316203272	1FXŲHCNPZ<	00000000000000000000		00000000000000000000	
27	54007177106323002100	= +THIS U	00000000000000000000		00000000000000000000	
30	54007176300610752100	= +¬XFH2Q	00000000000000000000		00000000000000000000	
31	66206010300612033414	VP≡ΠAFJC1L	00000000000000000000		00000000000000000000	
32	50140010340637040563	/L Π1F40E↑	00000000000000000000		00000000000000000000	
33	30053201340530070100	XEZA1EXGA	00000000000000000000		00000000000000000000	
34	70030100721754007231	+CA <U= <Y	00000000000000000000		00000000000000000000	
35	14056001300342013402	LE≡AXCZA1B	00000000000000000000		00000000000000000000	
36	20007230310160033405	P <∧YA≡C1E	00000000000000000000		00000000000000000000	
37	30071003160661707321	XGHLNF[+>U	00000000000000000000		00000000000000000000	
40	50007324137733105400	/ >ĮK(OH=	00000000000000000000		00000000000000000000	
41	73240306370207443101	>TCF4BG9YA	00000000000000000000		00000000000000000000	
42	60033405306310711251	≡C1ĘXCH+J(	00000000000000000000		00000000000000000000	
43	11500566300610060763	1/E∨XFHFG↑	00000000000000000000		00000000000000000000	
44	30071003160560103010	XGHLNE≡HXH	00000000000000000000		00000000000000000000	
45	12340554300710031606	J1E=XGHCNF	00000000000000000000		00000000000000000000	
46	60101400340450047321	≡HL 1U/D>U	00000000000000000000		00000000000000000000	
47	53040010054136041104	$U ΠE63D1D	00000000000000000000		00000000000000000000	
50	05703005320134053007	E+XEZA1EXU	00000000000000000000		00000000000000000000	
51	01007217000000000000	A <U	00000000000000000000		00000000000000000000	
52	00000000010073262000	A >VP	00000000000000000000		00000000000000000000	
53	11130200131450001100	1KB KL/ 1	00000000000000000000		00000000000000000000	
54	05100200120316020320	EHB JCNBCP	00000000000000000000		00000000000000000000	
55	14020100131002006305	LBA KHB ↑E	00000000000000000000		00000000000000000000	
56	04720200635004061400	U<B ↑/UFL	00000000000000000000		00000000000000000000	
57	34571444010013103057	1.LYA KHX.	00000000000000000000		00000000000000000000	
60	60201701601056001312	≡PUA≡H, KJ	00000000000000000000		00000000000000000000	
61	30131204055130201277	XKJUE(XPJ↑	00000000000000000000		00000000000000000000	
62	34055100055160103010	1E↑ E(≡HXH	00000000000000000000		00000000000000000000	
63	10060704140601001310	HFGULFA KH	00000000000000000000		00000000000000000000	
64	02000547301410031620	B E*XLHCNP	00000000000000000000		00000000000000000000	
65	32725400446061063230	Z<= -FHIS	00000000000000000000		00000000000000000000	
66	21005400460530220504	W = -EXRED	00000000000000000000		00000000000000000000	
67	14040100131030213406	LOA KHXQ1F	00000000000000000000		00000000000000000000	
70	02004577300234273003	B +(XB1=XC	00000000000000000000		00000000000000000000	
71	17013430060336303727	UA1^FC3X4W	00000000000000000000		00000000000000000000	

97404700C

```
DUMPTK(TK=267)        RESEXVF                               DUMPTK - VER. 1      74/07/19. 11.40.15.   PAGE   2     (7)
 72    30213422140134233024  AQIMLAISXT    00000000000000000000                   00000000000000000000
 73    13771105342401007326  K1I=ITA >V    00000000000000000000                   00000000000000000000
 74    00000000000006366322        F3IH    00000000000000000000                   00000000000000000000
 75    03150010640443001063  CM M#DB HI    00000000000000000000                   00000000000000000000
 76    67545055200106466300  A=/ PAF-I     00000000000000000000                   00000000000000000000
 77    43400043000106410677  05 o AF6FI    00000000000000000000                   00000000000000000000

WORD   TK=267  SE=3   B1=4    B2=100      TK=267  SE=4    B1=5    B2=100      TK=267  SE=5   [B1=6    B2=0]    EOR
 0     01010207000000000000  AABG          01010206000000000000  AABF          01010206000000000000  AABF
 1     12170235010207010000  JOB2ABOA      12170234010206010000  JOB1ABFA      12170234010206010000  JOB1ABFA
 2     24102205050000000000  THREE         24271700000000000000  TWO           24271700000000000000  TWO
 3     00000000000000000004          D     00000000000000000002          B     00000000000000000002          B
 4     00000000000000000000                24271733333400000000  T=0001        24271733333400000000  TW0001

WORD   TK=267  SE=36  B1=37   B2=100      TK=267  SE=37   B1=40   B2=0        TK=267  SE=40  [B1=0    B2=0]    EOF
                                                                  EOR
 0     00000000000000000000                00000000000000000000                00000000257600003242    U4  Z7
 1     00000000000000000000                00000000000000000000                01532205230530260600  A$RESEXVF
 2     00000000000000000000                00000000000000000000                00240002766500000311   T B-= C1
 3     00000000000000000000                00000000000000000000                17430010000001011245  OB H  AAJ+
 4     00000000000000000000                00000000000000000000                00323313000020010000   ZOK  PA
 5     00000000000000000000                00000000000000000000                25760000217600003002  U4  O-  XB
 6     00000000000000000000                00000000000000000000                00010100100000030600   AA H  CF
 7     00000000000000000000                00000000000000000000                30503051305214771701  X/X(X)LIOA
 10    00000000000000000000                00000000000000000000                05763075605030511237  F-X2E/X(J4
 11    00000000000000000000                00000000000000000000                10073474305113771006  HG1SX(K#HF
 12    00000000000000000000                00000000000000000000                33500462100602000135  0/O)MFB A2
 13    00000000000000000000                00000000000000000000                02000424011500055400  B OTAM E=
 14    00000000000000000000                00000000000000000000                01333001101431026114  AOXAHLYB(L
 15    00000000000000000000                00000000000000000000                15420100053610145400  M7A EJHL=
 16    00000000000000000000                00000000000000000000                03261063230023005400  CVHIS 5 =
 17    00000000000000000000                00000000000000000000                03252001301402000335  CUPAXLB C2
 20    00000000000000000000                00000000000000000000                05522002140102000335  EIPBLAB C2
 21    00000000000000000000                00000000000000000000                05041404617077745400  EOL#(+IS=
 22    00000000000000000000                00000000000000000000                02721702341750000533  B<OB1O/ EO
 23    00000000000000000000                00000000000000000000                54000275500001355400  = B2/ A2=
 24    00000000000000000000                00000000000000000000                03015000032554000305  CA/ CU= CE
 25    00000000000000000000                00000000000000000000                50000326540003061422  / CU= CFLR
 26    00000000000000000000                00000000000000000000                02000364307600030200  B C#X-=CB
 27    00000000000000000000                00000000000000000000                05475600057603132000  E=, E-CKP
 30    00000000000000000000                00000000000000000000                04773517300710060607  DI2OXGHFFG
 31    00000000000000000000                00000000000000000000                10713406140034070200  H+IFL 1GH
 32    00000000000000000000                00000000000000000000                06064017341650170001  FF5OIN/O A
 33    00000000000000000000                00000000000000000000                34033017020006161057  1CXOB FNM.
 34    00000000000000000000                00000000000000000000                34014017340730164417  1A5O1GXN9O
 35    00000000000000000000                00000000000000000000                36174017107110213303  3O5OM+HQ0C
 36    00000000000000000000                00000000000000000000                44170605300105030100  9OFEXAECA
 37    00000000000000000000                00000000000000000000                02213004020004462000  BQXOB O+P
 40    00000000000000000000                00000000000000000000                13653415200013015400  K#1MP KA=
 41    00000000000000000000                00000000000000000000                05332000053654000135  EOP EJ= A2
 42    00000000000000000000                00000000000000000000                14002310362001010134  L 5H3PAAA1
 43    00000000000000000000                00000000000000000000                30143502106335011014  XL2BH:2AHL
 44    00000000000000000000                00000000000000000000                31026010301113771006  YB=HX(K#HF
 45    00000000000000000000                00000000000000000000                33100410230435030561  OHOHSD2CE(
 46    00000000000000000000                00000000000000000000                30120502301534150100  XJEBXM1MA
 47    00000000000000000000                00000000000000000000                01525400031110636010  A)= C1H!=H
 50    00000000000000000000                00000000000000000000                30113402301034010346  X11BXH1AC-
 51    00000000000000000000                00000000000000000000                00000001163500000000     AN2
 52    00000000000000000000                00000000000000000000                14771701057630766010  LIOAE-X-=H
```

FILE MANAGERS 10

10.0 INTRODUCTION

This section describes the two KRONOS 2.1 file managers: Permanent Files and the Local File Manager.

10.1 PERMANENT FILES

Permanent files are controlled by the system PPU routine, PFM (Permanent File Manager). All requests for permanent file action are accompanied with a specific user number. User numbers are established by installation personnel and entered into the system validation file, VALIDUX. Thus, only users known to the system may request permanent file action.

There are two types of permanent files available to users of KRONOS: direct and indirect access files.

- A direct access permanent file is read and written by user I/O requests just as any local file would be read or written. Large data files occupy large amounts of mass storage and are normally created as direct access files.

- An indirect permanent file is accessed by using a working copy of the file rather than the file itself. The working copy is attached as a local file to the user job. Thus, modifying the working copy does not alter the actual permanent file. Indirect access files are allocated in 64 CM word blocks and are generally used for small permanent files.

A direct access permanent file is normally declared by the user prior to writing the file by using the DEFINE control card or macro. However, this control card may be used after the file is written, if desired. Indirect access permanent files are declared by the SAVE control card or macro after the file has been written.

Whenever a permanent file is declared, the user number is mapped into a CATALOG track where permanent file names and statistics for that user are maintained. Thus, there is one catalog entry for every permanent file known to the system. A catalog track normally contains entries for several different users. A description of this mapping is provided in Section 4 of the KRONOS 2.1 Installation Handbook.

A family consists of 1 to 63 mass storage devices. Within a family, each user has a master device that contains his permanent file catalogs, all of his indirect access files, and some or all of his direct access files. Again, the mapping of a user index into a master within the family is shown in Section 4 of the KRONOS 2.1 Installation Handbook.

If more than one family is available in the system, the user must specify which family via the ACCOUNT control card.

A user may specify a list of other users permitted to access his permanent files. This list is specified on the PERMIT control card or macro, and results in adding an entry to the PERMIT buffer.

PFM is the permanent file manager routine. It is called to a PP either by an RA+1 call generated by the user CP Macros, by a control card call, or by TELEX.

1. A CP programmer may use any of the PFM calls described in the reference manual. These calls will produce an RA+1 call to PFM.
2. Any PFM control card will cause the CP routine PFILES to be loaded in the user FL. PFILES will issue the PFM macro calls which result in an RA+1 request for PFM.
3. A user on a TTY can issue PFM requests with the proper TELEX time sharing commands. With the exceptions of CATLIST, TELEX will call PFM directly with an RA+1 request. The terminal user does not need to be sent to a CP. The CATLIST command however, must be issued from a CP since the call block will not fit into one POT. TELEX will issue a dummy CP to issue the CATLIST call to PFM.

As a review, the two types of PFs are shown below:

PERMANENT FILES

DIRECT	INDIRECT
LARGE files Track allocation single copy write interlock multi-read (multiple FNT/FST)	SMALL files sector allocation multi-copy each user gets his own copy of the file.
DEFINE(lfn=pfn) ATTACH(lfn=pfn)	SAVE, pfn GET,pfn REPLACE,pfn
Fast Attach capability from the macro ATTACH pfn,,,,,,,FA and ISF. ISF(R=)	

PERMISSIONS:

Read,Execute only, Write, Append, Modify
either explicitly
or implicitly

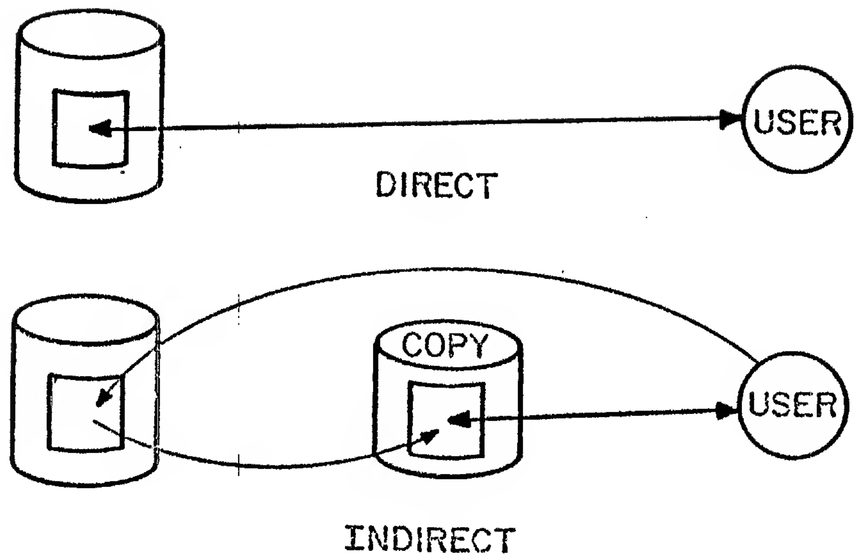


Figure 10-1.1 Permanent Files

	TYPE
implicit permissions	PUBLIC, SEMI-PUBLIC
explicit permissions	PRIVATE

requires

PERMIT(pfn,usernum=mode,PN=packname,R=r,NA)

10.1-1 Master User of PFs.

UN where "*" occurs can access any PF in read mode for an UN whose after characters correspond to the master user.

For example, UN ABC* can access all the PFS of UN ABC1, ABC2, ABCX, in fact any 4 character UN whose first three characters are ABC. When a user requests PF activity PFM will use the users User Index, UI, to map to that particular users Master device and catalog track. The algorithm for this mapping is in the Installation Handbook part IV.

Each Master Device, MD, has a predetermined number of catalog tracks. The MST DEVL word 4 contains PF information.

DEVL	IAPF	Label Track	Permit Track	Number Cat Tracks	System Table Track

Byte 3 contains the actual number of tracks which are catalog tracks. They must be contiguous tracks. Byte 1 is the first track of the label track chain. This track chain consists of the label and all the catalog tracks. Normally the label track is track 0, however, if track 0 is flawed, then the first available track will be used. The d bit is set in the TRT so that this track chain is preserved across deadstarts. Normally, then, track 1 will be the first track of the catalog tracks. Only MDs have catalog tracks, so for a non MD device the Label track chain consists of only 1 track.

However, if all the catalog tracks cannot be contiguous starting at track 1, then the first available contiguous tracks large enough for the catalog will be used. In this case bit 11 is not set in the PFDL word MST+6, and the 1st catalog track is pointed to by the TRT link from the label track position.

Since each user is mapped to a particular catalog track, and these tracks are contiguous, the link byte in the last sector does not link to the next track in the chain. If this track becomes full, catalogs cannot overflow to the next contiguous track (other users are mapped to that track). A new track is linked into the chain via the TRT table, and the last sector link byte

points to this new track. Effectively then, PFM has just increased the length of this catalog track. This, of course, will slow down PFM when he has to search more than 1 catalog track for any one user. So bit 10 is set in PFDL and an 0 is displayed in the E, M display.

Many users can be mapped to any specific catalog track, but no user can be mapped to more than one catalog track (in case of overflow, it is considered a very long track).

As a user creates PFs an entry (Figure 10-1) is placed in the appropriate catalog track and the file is processed by PFM.

If the PF is direct access file DPF, then 1) If the file resides on a device in the users family which can contain PFs the entry is created and the first track is recorded, the first sector entry is set to 4000B denoting a direct access PF. Since this is a regular file, sector 0 will contain the system sector, and Sector 1 will be the first sector of data. PFM will issue the STBM function to set the d bit in the TRT. 2) If the file does not reside on such a device, the job is aborted unless error processing was desired. In order to avoid this possibility, the user should DEFINE the file prior to writing on it.

If the PF is indirect PF IFP, then the entry will be copied from the regular file which is to be made permanent; i.e., DPF are regular files which have the d bit set in the TRT. IFP are not kept as regular but are allocated by PFM, and the system does not keep track of them. The user must create the file first, and then issue the SAVE command.

PFM keeps an IPF track chain. This chain is reserved from the system as a normal file chain, the d bit is set to preserve it over deadstarts. Word DEVL byte 0 points to the first track of this chain. The chain is kept to a minimum length when possible, and is expanded RTCM and contracted DTKM or DLKM as necessary. However, the IPF track chain must completely reside on its MD since every user mapped to the MD must have all his IPFs on this device. The format for the file is shown below. Note that Sector 0 is the system sector of the IAPF and sector 1 is an eoi, Hence Sector 2 contains the 1st data which is the 1st IPF saved.

0	1	n	n + 1			
SS	eoi	1st file	eoi	2nd file	eoi	etc

As each SAVE command is processed, PFM will get n contiguous sectors (the length of the users file not counting the system sector) on the IPF chain. It will copy the users file exclusive of the system sector, but including the EOI. The number of sectors copied, not counting the EOI sector is saved in the catalog entry as well as the first track and sector number of the file. Sector 4000B does not exist so there is no confusion between DPF and IPF entries in the catalog entries.

As more files are SAVEd and DEFINEd the catalog entries grow and grow and could cause overflow as we have seen described earlier. However, available slots in the catalog entries are created by PURGEs of PFs. These available slots are known as holes.

When a DPF is purged, the UI is set to zero, the number of sectors is set to zero, and all the tracks in that file chain are released to the system. This hole can be used for new DPFs or new IFPs.

When an IFP is purged, its UI is set to zero, however, the sector count field is left intact. The sectors are not released physically unless the file was so large it spanned one or more whole tracks. In which case, the tracks are returned to the system DLKM, and the sector count field is set to the remaining sectors. This hole can only be used by new IPFs.

In the case of the REPLACE command, the following occurs:

1. If the new file is the same size as the existing file, the new file is copied over the old IFP file.
2. If the new file is smaller than the existing file, the new file is copied over the the old one, the sector count field is modified, and a new PF catalog entry is built. This entry has UI=0, and sector count field set to the remaining sectors, and first track and sector pointing to the remainder of the old file.
3. If the new file is larger than the old file, the current entry is set to a hole, UI=0. A new hole is found if one big enough exists, or the new file is placed on the end of the IAPF, and a DPF hole or a new catalog entry is used.

Hole searching is accomplished the same way for both a SAVE and a REPLACE command. Only the catalog track (plus overflow tracks, if any) mapped to by the UI of the user are searched; i.e., the entire catalog track is never completely searched.

1. If a hole with the exact number of sectors available is found, it is used.
2. If not 1, then the largest hole, larger than the file is used.
3. If not 1 or 2, then the file is put on the end of IAPF and a new entry or a DPF catalog is used.

The scheme of searching for largest residue holes makes the best utilization of the IAPF, since very small holes seldom get filled. Eventually, of course, the IAPF gets very holey, and a PFDUMP and PFLOAD are the only solution. PFLOAD will recreate the catalog entries and IAPF with no holes.

When a GET command is issued, PFM will find the entry, and copy the file from the IAPF to a local file and create an FNT/FST entry for this local file with the proper permission bits set. PFM counts the sectors copied (exclusive of the eoi) and compares them with the sector count field, and if they do not agree it will issue a file sequence error.

When an ATTACH command is issued, PFM will find the catalog entry and create an FNT/FST pointing to this file. The file type is PMFT and the permission bits are set accordingly.

Of course on any GET or ATTACH command, PFM will be mapped to the proper catalog track and will ensure that the user has either explicit or implicit permission to use the file.

In the case of implicit permission the file is made available.

In the case of explicit permission, the catalog entry points to a permit track, where the permission entry for this file resides. Byte 2 of DEVL points to the first such track. This entry, figure 10-2, indicates the permissions available to this user.

In all cases, the original owner of the file always gets all permissions when he or she requests it from PFM.

Finally, PFM always searches the FAFT type entries in the FNT/FST first on any ATTACH function. Then PFM searches the appropriate catalog track.

In the case of an FAFT file, PFM knows the family of the requestor, and can return the proper file in the case of duplicate PF names in different families. The FAFT requestor is returned a PMFT type FNT/FST entry. This just allows an FAFT file to be found quickly, hence the name Fast Attach File.

Formats of the catalog entry and PERMIT buffer are shown in Figures 10-1 and 10-2.

Word	59–48	47–36	35–24	23–18	17–12	11–0
0	Permanent File Name					User Index
1	File length in PRU's			First Track		A / First Sector
2	random index		Creation Date and Time			
3	access count		Modification Date and Time			
4	ct / mode	/ dn	Last Access Date and Time			
5						
6	Optional Password					
7	Program Control Word					

Figure 10-1. Catalog Entry Format

where:

A	-	if FIRST SECTOR = 4000B this is a DPF
random index	-	the random disk address of PERMIT sector
Access count	-	the number of times this file was accessed
ct	-	file category as follows:

0 = private
1 = semiprivate
2 = library or public

mode	-	Mode of access for semiprivate and public files as follows:

0 = write, read, execute, append, modify, and/or purge
1 = read and/of execute
2 = append
3 = execute
4 = negate previous permission
5 = modify
6 = read and/or execute, allow modify
7 = read and/or execute, allow append

Figure 10-1. Catalog Entry Format (Continued)

dn - Device file resides on (0-77B). If 0, file resides on master device. dn ≠ 0 for a direct access file residing on a device other than the master.

date/time - All date and time entries are in octal with the following format:

yymmddhhmmss

yy is biased by 70.

program control word - User control information from FET+11D.

Figure 10-1. Catalog Entry Format (Continued)

	59 — 36	35 — 18	17 — 0	
0	Random index	User index		used for linkage
1		date and time		
2	Permitted user number		User index	up to 31 PERMIT entries (2 words per entry, >31 overflow to next sector)
3	Access count Mode	Access time	Access date	
		⋮		
77B				

where:

random index - Disk address of next PERMIT buffer for this user index. Zero indicates end of chain.

user index - User index who created this sector.

Access count - The number of times the permitted user has accessed the file.

Mode - Mode of permission given to user

Access Time/Date - Time and date of last access by permitted user.

Figure 10-2. PERMIT Buffer

The permanent file manager can be called in two ways: a system call or a TELEX call. The call formats and call blocks are different for the two calls as shown in Figures 10-3 through 10-6.

- System call initiated from the CP routine PFILES, or a system call is from a macro.

IR	PFM	CP#	CC		FET
	18	6	12	6	18

where:

FET - Address of the 14-word call block

CC - Command code (request) as follows:

Symbol	Value	Command
CCSV	01	SAVE
CCGT	02	GET
CCPG	03	PURGE
CCCT	04	CATLIST
CCPM	05	PERMIT
CCRP	06	REPLACE
CCAP	07	APPEND
CCDF	10	DEFINE
CCAT	11	ATTACH
CCCG	12	CHANGE catalog data

Figure 10-3. System Call Format

The 14-word call block pointed to by FET is shown in Figure 10-4.

	59 … 18	17 … 0	
FET+0	logical file name (lfn)	STATUS	
	dt | | E P |	FIRST	
		IN	
		OUT	
		LIMIT	
FET+8	Permanent file name (pfn)	ct mode	CFPN
+9	Optional user number (ovan) ⟷	np →	CFOU
+10	Optional file password	erad	CFPW
+11	Program control word (pcw) SS		CFUC
+12	pack name (pkn)	units	CFPK
+13	new file name (nfn)		CFNF

Bit positions: 59, 48 47, 36 35, 24 23, 18 17, 12 11, 0

Figure 10-4. System Call Block

where:

Status - Bit 0 must be zero prior to calling PFM.
Error codes are returned in bits 10-17.
Bit 0 is set to one upon completion of the request.

First - Buffer pointers are used by CATLIST function

dt - Device type of file residence

ep - Bit 44. If set, control is returned to the user on errors.

pfn - Permanent file name. If zero, lfn is used.

ct - File category (private, semi private, library).

mode - File access mode

ovan*1 - Alternate user number

np*1 - Number pf PRUs required for the direct access permanent file being DEFINE'd.

erad - Address where error messages are returned.
The message may be up to three words long and is stored at the given address only if ep is set.

pcw - Program control word. Whatever the user stores in this word is stored in the catalog entry when a permanent file is created. This word is read from the catalog entry and stored in CFVC when the file is attached.

SS - Sub-system designation set by TELEX when the file is being accessed via the time sharing executive.

pkn - Name of the auxiliary device to be used in satisfying the permanent file request.

units - The number of units of the type specified by dt. For example, if the device type is DI4, the dt field contains DI and the unit field contains 4.

nfn - New file name used with the CHANGE command.

*1 Mutually exclusive fields, FET may contain either but not both fields.

Figure 10-4. System Call Block (Continued)

- TELEX Call

A TELEX call is initiated by a call from the time-sharing executive. The call format is shown in Figure 10-5 while the call block is shown in Figure 10-6.

IR	PFM	CP	CC	TN	PP
	18	6	12	12	12

where:

CC - Command code as for system call.

TN - Terminal number used to index into the terminal table within TELEX.

PP - Pot pointer used to locate the call block (also within TELEX).

CP - CP number equals 1 indicates TELEX call when TELEX running.

Figure 10-5. TELEX Call Format

Word	59 … 18	17 … 12	11 … 0	
0	logical file name	STATUS		TXSN
1	file name table (FNT) entry			TXFT
2	file status table (FST) entry			TXFS
3	permanent file name	ct	mode	TXPN
4	optional user number			TXOU
5	file password	rclad		TXPW
6	program control word			TXUC
7	packname	eq	ctls	TXPK

where:

rclad - Address of input register if recall needed.

eq - Equipment in family to be accessed.

ctls - User control bits

Error messages are returned in words 0-4 of TELEX call block.

Figure 10-6. TELEX Call Block

Routines called by PFM include the following:

- 0AV - account verification
- 0BF - begin file
- 0DF - drop file

The important thing to remember here is that the lengths of these three routines are defined as assembly constants in PFM. Thus, any change in their lengths might affect their loading in PFM.

PFM consists of a few resident subroutines and the following overlays:

3PA	-	Command processor
3PB	-	SAVE, REPLACE, APPEND processors
3PC	-	APPEND processing
3PD	-	ATTACH processing
3PE	-	Catalog list routines
3PF	-	DEFINE processing
3PG	-	PERMIT/PURGE processing
3PH	-	Error processing
3PI	-	Auxiliary routines
3PJ	-	CHANGE processing

There are four addresses where the 3P(x) overlays are loaded. One address, OVLA, is defined in the main routine, PFM. Two are defined in overlay 3PA. They are BUF and OVLC. The last address, BFMS, is defined in SYSTEXT. Figure 10-7 shows the 3P(x) overlays, their load addresses, and their approximate lengths. Overlays loaded at OVLA should not extend beyond BFMS, while those loaded at OVLC should not extend beyond OVLL. There is enough space between OVLL and OVLC to allow for one full PRU, plus one short PRU. (this is the concern of the PP resident loader).

Not shown in Figure 10-7 are the load addresses for the 0-level overlays. These overlays include 0BF, 0DF, and 0AV as mentioned earlier. However, these routines have been taken into consideration in that their lengths have been included in the 3P(x) overlay lengths shown in Figure 10-7. Furthermore, the load addresses for the 0-level overlays are referred to by the symbol, LOCF, in the 3P(x) overlays and, as such, are quickly located with the aid of the symbolic reference table when looking at the listing of PFM.

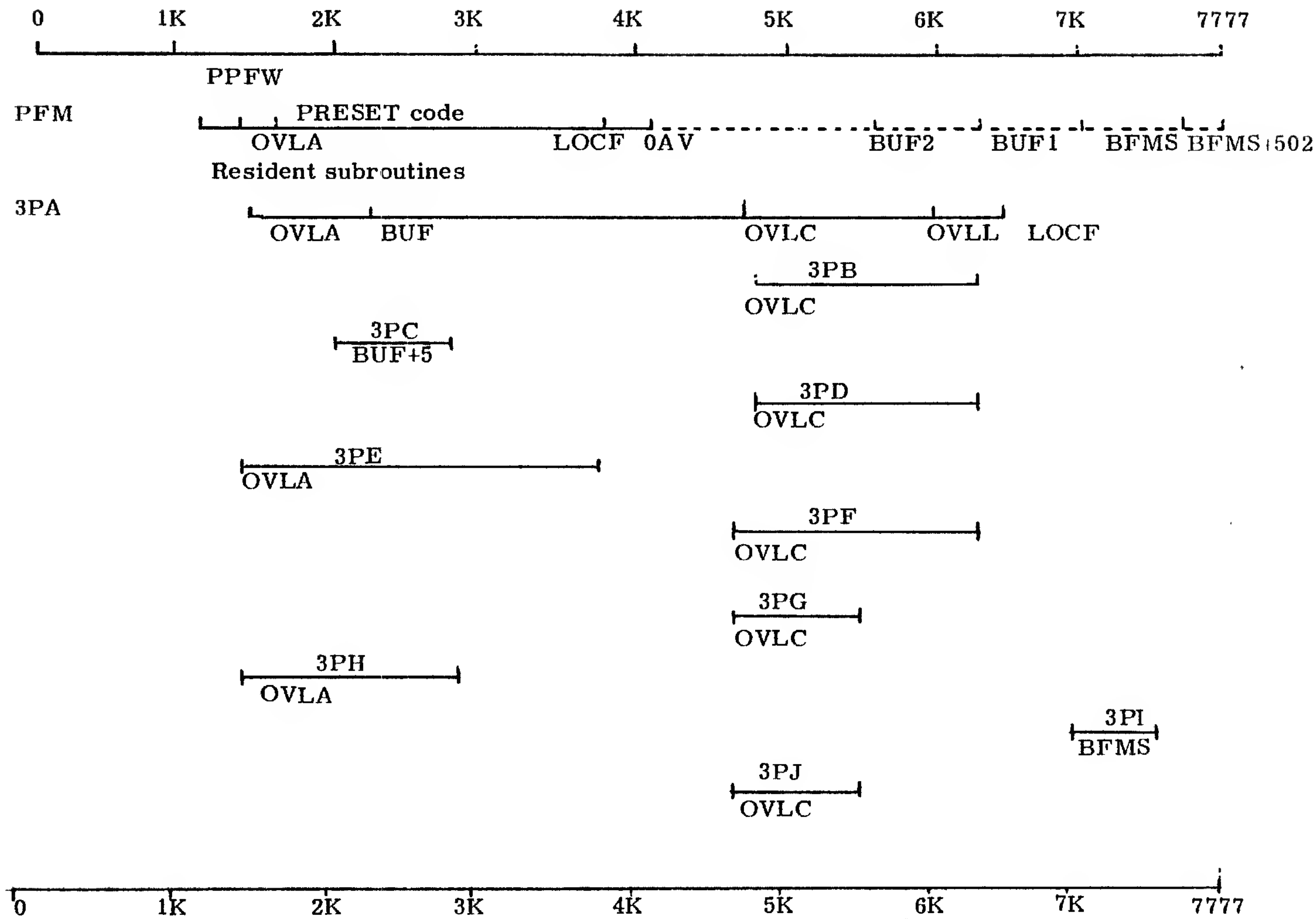


Figure 10-7. PFM Overlay Load Map

10.1.1 PFM - Permanent File Manager

PFM provides storage for the call block and other temporaries. Its resident subroutines are:

- SFA - Set FET Address
- CCI - Clear Catalog Interlock
- DPP - Drop PPU
- TTA - Set Terminal Table Address
- ERR - Process Error
- SFN - Set File Name in CM
- MSR - Mass Storage READ Error Processor

PFM presets processing routines to:

- Verify FET parameters
- Verify user validation allowances
- Place request in recall if catalog is interlocked
- Issue accounting messages
- Load proper function processor overlay (3PA or 3PE)
- Call RESEX if pack is unavailable.

10.1.2 3PA - Main Command Processing

3PA performs all processing required to perform the GET function. 3PA performs preliminary processing for the following commands:

SAVE	PERMIT	CATLIST
ATTACH	APPEND	REPLACE
PURGE	DEFINE	CHANGE

3PA also contains:

- Catalog processing routines
- PERMIT processing routines (some)
- File allocation routines
- General subroutines
- Device-to-device transfer routines

The following outline describes 3PA subroutines and buffers in more detail. Many of the 3PA subroutines are called from other 3P(x) overlays. Those called from another overlay are labeled with an asterisk.

- 3PA resident routines include:

  * TRP - terminate program
  TST - terminate TELEX request
  * EOI - write end-of-information

- Resident common decks include:

  COMPSNT - set next track
  * COMPRNS - read next sector

- Device-to-device transfer routines include:

  * DTD - main routine
  PTE - process transfer error
  * IBA - increment buffer address
  * SDP - swap disk parameters
  * WNS - write next sector

- * BUF - device-to-device transfer buffer overlays subsequent subroutines

- PERMIT subroutines include:

  * CPE - create PERMIT entry
  * CPI - check permission information
  * UPI - update permission information
  CSA - compute sector address
  SPI - search permission information
  PPE - process PERMIT read error
  WNP - write new PERMIT buffer
  FPE - form PERMIT entry in buffer

- Catalog processing subroutines include:

  * CCS - create catalog sector
  * DCE - delete catalog entry
  FCE - form catalog entry
  FHE - form hole entry
  * UCE - update catalog entry
  * SSC - select catalog entry

* SCH - search catalog
PCE - catalog READ error processor
CCD - check catalog data

- Allocation subroutines include:

  AFS - allocate file space for indirect file
  ACS - allocate catalog space
  APS - allocate PERMIT space

- General subroutines include:

  * DIK - drop tracks
  * ITC - interlock track chain
  RTK - request linked track
  * WBI - write buffer in place
  COMPCRA - convert random address
  * COMPSEI - search for end-of-information
  * COMPCTI - clear track interlock
  * COMPSTI - set track interlock
  * COMPCKP - set checkpoint bit in EST entry

- OVLC - command processing overlays are loaded here and destroy the following subroutines. These overlays must not exceed OVLL.

- GET and ATTACH processing routine
- Command processing initialization - SET
- Catalog search initialization subroutines include:

  * ISP - initialize search
  * SPN - set permanent file name
  * COMPSAF - search for assigned file
  * COMPSFB - set file busy

3PA calls many of the other 3P(x) overlays.

Those called are shown in Table 10-1.

TABLE 10-1. OVERLAYS 3P(x) CALLED BY 3PA

Overlay Name	Load Address	3PA Subroutine Called From	Command Processed
3PI	BFMS	ISP	
3PB	OVLC	SET	SAVE/APPEND/REPLACE
3PJ	OVLC	SET	CHANGE
3PF	OVLC	SET	DEFINE
3PF	OVLC	SET	PURGE/PERMIT
3PD	OVLC	GET	ATTACH

10.1.3 3PB - SAVE/REPLACE/APPEND Processing

The 3PB overlay contains subroutines for processing the commands: SAVE, REPLACE, and APPEND. It also contains some common subroutines. An outline of the subroutines comprising 3PB is given below:

- APP - process APPEND command
- REP - process REPLACE command
- SAV - process SAVE command
- Subroutines include:

  CUC - check user controls

  PFR - process file replacement

  SSP - set statistical parameters

  SSF - search for system file

  PRS - preset 3PB

Only one overlay, 3PC, is called by 3PB from subroutine APP.

10.1.4 3PC - APPEND Processor

Overlay 3PC is loaded at BUF+5 by subroutine APP in overlay 3PB to process the APPEND command. The order of the transfer is as follows:

- the old permanent file is copied to a new permanent file, then
- the local file is copied to the new permanent file.

10.1.5 3PD - ATTACH Processor

Overlay 3PD is called from subroutine GET in overlay 3PA to process the request to attach a direct access file to a job. 3PD consists of the following subroutines:

- ATT - process ATTACH command main program
- CFM - check file mode
- Common decks:

COMPSDI - set P.F. device interlock

COMPRSS - read system sector

COMPWSS - write system sector

- MSS - read system sector error processor
- CFA - fast attach file processing
- COMPFAT- search for fast attach file

Subroutine ATT calls 0BF for fast attach files.

10.1.6 3PE - Catalog List Routines

Overlay 3PE is called from the preset subroutine, PRS, in the main program, PFM. 3PE is loaded at OVLA and is called to read permanent file catalogs for a central processor program. Data is returned to the CM buffer specified by the FET pointers: FIRST, IN, OUT, and LIMIT. The PFM call format for this request is:

RA + 1	PFM	P*	CCCT	0	FET
	18	6	12	6	18

* Recall Bit if desired.

The call block pointed to by FET is shown in Figure 10-8.

	59 18	17 0
FET+0	logical file name	Status
		FIRST
		IN
		OUT
		LIMIT
FET+6	Reserved	
	permanent file name	0 mode
	ovan	

where,

status = 33 if buffer is full

= 1033 if request completed (buffer is filled from IN to LIMIT - 1).

FET+6 - Reserved for recall information to PFM

Mode = 0 to search catalog entries

≠ 0 to search permit entries

Ovan - Alternate user number thus search alternate catalog. In this case, the password and user index are cleared before being written to CM buffer.

Figure 10-8. PFM Call Block

Overlay 3PE consists of the following subroutines:

- CAT - main program
- NCS - normal catalog search (mode=0)
- ACS - alternate catalog search
- PDS - PERMIT data search
- SBS - set status of FET (update IN)
- RBS - read buffer for search
- SHB - search catalog buffer
- WDB - write buffer
- CCP - check catalog permission (clear password)
- DFS - determine file size (store in catalog entry)
- SPB - PERMIT buffer search
- CSA - compute sector address
- Common decks include:

  COMPCRA - convert random address

  COMPSRA - set random address

  COMPSEI - search for end of information

  COMPRNS - read next sector

  COMPSDN - search for device number

- Buffers BUFA and BUFB overlay following code
- CSU - check for special user
- ISP - initialize search of catalog with

  COMPSCA - set catalog address

10.1.7 3PF - DEFINE Processor

Overlay 3PF is called to create a direct access permanent file. The file exists prior to the DEFINE command, or the file may be created after the DEFINE command. File residency is determined in 3PF for the two situations as follows:

- Local file exists - the local file is made permanent if the local file resides on a PF device; otherwise, the request is aborted. The dt field of the call block is ignored. If the local file resides on a removable device, that device's packname must be the same as the packname specified in the call block.

- No local file - If the dt field is zero, the file is placed on the device with the most available space. If dt is specified, the file is placed (started) on the device of that type with the most available space. If np (number of PRUs) is specified, the file is placed on the device (type dt, if specified) with the most available space, provided that np PRUs are available. If np PRUs are not available, the request is aborted with the message: "PRUS REQUESTED UNAVAILABLE".

3PF writes a system sector for the file to reflect the permanent file status of the file. The catalog entry is stored in the system sector as indicated in the format below. However, note that byte 2 of word 1 is updated to indicate the current access modes. Also, word CTSP may contain the catalog pointers. However, there are no references to this word in PFM, so it may be assumed that the word is unused by PFM. Nevertheless, the format of the system sector is shown in Figure 10-9.

Word	59–48	47–36	35–24	23–12	11–0	
0	logical file name					
1	EQSS	FTSS				
2			date and time			
3						
4						
5						
6						
7						
10	Word 0 of catalog entry					CTSS
11	0		0 | am	0		
	Words 2-7 of Catalog Entry					
21						CTSP
22	0		rm | ra	r	0	CTUC
77	unused					

where,

EQSS - Equipment number of system sector

FTSS - First track

am - Current access modes as follows:

bit	meaning
24	file currently attached in read mode
25	file currently attached in write mode
26	not used
27	file currently being modified or may be modified
28	file currently being extended or may be
29	file purged

Figure 10-9. System Sector Format

rm	-	Number of users with RM or M access set to zero by 3PF.
ra	-	Number of users with RA or A access. Set to zero by 3PF.
r	-	Number of users with R or W access. Set to 1 by 3PF.

Figure 10-9. System Sector Format (Continued)

Overlay 3PF consists of the following subroutines:

- DEF - main routine to build catalog entry write system sector.
- CUC - check for maximum number of files reached.
- DFR - determine file residency
- CPR - check for proper family or pack name residency.
- DDN - determine device name from MST entry.

10.1.8 3PH - Error Processor

Overlay 3PH contains the error processing routines for all other overlays. It performs the following:

- Sends the indicated error message to the dayfile.
- Sets the FST entry "not busy", or
- Deletes the FNT/FST entry if created by PFM
- Terminates the calling program if user error processing is not specified.
- Drops the PPU
- If a TELEX call: returns error message in a POT, sets the completion bit, and drops the PPU.

Overlay 3PH contains a list of error messages issued by PFM. This list is available in the KRONOS 2.1 Reference Manual. Some messages are sent to the control point dayfile while others are sent to the error log.

10.1.9 3PI - Auxiliary Routines

Overlay 3PI contains auxiliary routines used by many of the other 3P(x) overlays. These auxiliary routines can be overlayed after execution by any process that uses BFMS since 3PI is loaded at BFMS. Currently, 3PI contains two common decks:

COMPSCA	-	set catalog address, and
COMPSDN	-	search for device number

3PI must not extend beyond BFMS+502.

10.1.10 3PJ - CHANGE Processor

Overlay 3PJ processes the CHANGE command by changing and replacing the catalog entry for a file.

10.1.11 SAVE Command Processing Flowchart

The flowchart Figure 10-10 represents processing for the SAVE command. Subroutine SAV is contained in overlay 3PB which has been called from 3PA. Notice that control returns to 3PA by entering subroutine DTD. Subroutines SSC and CCS are also contained in overlay 3PA.

10.2 LOCAL FILE MANAGER

Local file management consists of a set of macros, control cards, and the PP routine (LFM). The common decks required for the macros processed by LFM are COMCLFM and COMCSYS. LFM performs various file managing functions for a job. A description of each function and its macro call is available in Section 7 of the KRONOS 2.1 Reference Manual. The PP program LFM consists of a group of overlays that perform the requested function. The functions and their corresponding LFM overlays are outlined in Table 10-2.

TABLE 10-2. LFM OVERLAYS

Code	Function	Overlay	Entry Point
0	Rename file	3LB	RNM
1	Assign Common file	3LD	ACF
2	Enter Common file	3LD	ECF
3	Release common file	3LD	RCF
4	Release print file	3LE	RPR
5	Release punch file	3LE	RPH
6	Release PUNCHB file	3LE	RPB
7	Release P8 file	3LE	RP8
10	Lock file	3LB	LCK
11	Unlock file	3LB	ULK
12	Return file status	3LB	RLS
13	Return current position	3LB	RCP
14	Request equipment	LFM	RQI
15	Assign equipment	LFM	AEI
16	Release files	3LE	REL
17	Set file ID code	3LE	SID

10.2.1 Local and Other Files

A File is a collection of data saved on a storage medium. It can be tape or mass storage. The data is written in groups of blocks or sectors, as has been shown in chapter 7.

The system controls and designates a file by its File Name Table FNT and keeps its position by the File Status Table FST.

There are basically two kinds of files.

1. Explicit files defined by an FNT/FST.
2. Implicit files that are known only by a track reservation in the TRT. They are actually track chains and, as such, are managed by the owner. They are, more specifically, files unknown to the system. The best example is the Indirect PF track chain; see PFM in this chapter.

This discussion will concern itself with only those files known to the system explicitly.

These files all have an FNT/FST entry. The FST is basically used for file positioning information, with exceptions for Queue type files. The FNT is shown below

	42	6	6	1	5
FNT=	lfn	job org	type		CP number

If system sector present, bit 5 ~~is on.~~

The FNT/FST is created for a variety of reasons. With the exception of 1TA for TELEX rollout files only, all files are created (i.e., FNT/FST entry built) by the PP routine 0BF begin file. With no exceptions, FNT/FST entries are cleared (i.e., files dropped) by the PP routine 0DF dropfiles, with help from 0FA to release FA files, and 0PR to release DPF files and 0RF to update RESEXDF and RESEXVF for non-allocatable files, tapes and removable packs.

An FNT entry is considered empty if the lfn=0. When a new file is created, an empty entry is found and used for this new file. See the discussion on the pseudo channel FECT in Chapter 2. 0BF will create a file with any lfn, even those consisting of special characters. Only a PP routine can use 0BF, so a PP routine can create a file with any 1 to 7 character name. CP users, however, must ask CIO to create a file entry for them. CIO requires that a name be legal. A legal file name is composed of 1 to 7 alphanumeric characters. If CIO finds a special character in a file name which is to be created, he will abort the CP. However, CIO

will accept a file which has previously been created with an illegal name for reading, writing, or positioning. This allows a DMP=SEP routine to use the file DM*, which was created by 1RO. Once CIO has determined that the lfn is legal, it will call 0BF to create the FNT/FST entry.

The job origin field will always contain the origin code of the creator of the file, SYOT=0, BCOT=1, etc.

Bit 5 is always set for RMS files, since all RMS files must contain a system sector.

The CP number field contains the CP number of the current user of the file. If it is set to zero, then the file is in a Queue.

The type field defines what type of file it is. The types will be discussed individually below. Refer to the example of an FNT in Chapter 2 for the following discussions.

The FNT size can be specified at deadstart time in the CMRDECK. The default size is 1000B, which allows up to 400B files to be active in the system simultaneously. The entries are each 2 words long and are numbered. These numbers are known as the FNT ordinal. The first FNT is ordinal 0 and is always the file SYSTEM. FNT ordinal 1,2,3, and 4 are always created at deadstart time and are respectively VALIDUX, SALVARE, RESEXDF, and RESEXVF. The first available FNT entry then is always FNT ordinal 5.

The type field is set up by 0BF in the following manner. A table of file names is kept in 0BFs FL. The file type is set to the corresponding file name. If the caller of 0BF desires that the file have a file type different than 0BF generates, the caller must change it himself. PFM will change the type to LOFT for GET command, in case the name was one of those in the OBF table, and will change type to PMFT or SYFT for ATTACH commands.

The table as of Level 5 is at TSFN in 0BF.

	lfn	type
TSFN	INPUT	INFT
	OUTPUT	PRFT
	PUNCH	PHFT
	PUNCHB	PHFT
	P8	PHFT
	LGO	LOFT
	any other name	LOFT

1. Type INFT=0, ROFT=1, PRFT=2, PHFT=3, TEFT=4 have been extensively described in chapters 5 and 6.

2. Type SYFT=5. System type files are files which are used by the system for special functions. The 3 most famous SYFT files are VALIDUX, RESEXDF and RESEXVF which are created by the deadstart procedures and permanently remain at FNT ordinal 1, 3, and 4 respectively.

   These file types are changed to FAFT whenever ISF is run at a CP. If the ISF (R=lfn) is used, then the lfn specified if type FAFT will be changed to type SYFT, or else the type remains unchanged. See FAFT below.

   One other file is made SYFT if defined by PROFILE and that is the PROFILO file.

3. LOFT=6. Local type files are generally scratch files. They are any file created locally at a CP and any indirect file retrieved by the GET command. These files are automatically released by 1CJ at job completion time. All tape files are also considered local files. See FNT ordinal 7.

4. CMFT=7. Common files are classed into two categories.

   a) Unlocked common files are denoted by the write lockout bit set off (bit 12 which is really the read only bit). See FNT ordinal 3.

      When a user wants the use of this file, and he is validated for common files, he issues the LFM command COMMON. If the file is in the Common Queue, the CP # field = 0 and LFM will set it to the CP# of the caller. The caller then can read, write, or release the file. Only one user may use this file at a time. If the CP# field is not zero, the user will have to wait until it becomes available. If the file does not previously exist, LFM will set the type field to CMFT. When the user returns the file or ends, the CP# field will be set to zero and the FNT/FST will not be cleared. When the user issues a RELEASE on a common file, LFM will set the type field to LOFT and at return or end of job, 1CJ will drop the file.

   b) Locked common files are denoted by the write lockout bit set (bit 12). The bit is set by the LOCK command and unset by the UNLOCK command. However, when the creator of the file returns it or drops, and if the write lockout bit is set, then the file can never be UNLOCKED or RELEASED, except by a level zero deadstart, or with the console memory entry commands.

      When a user wants the use of this type of file, he issues the COMMON command. LFM will find the FNT and will create a new FNT/FST for the user of type LIFT. This file will be in READ only mode. Many users can be reading this file

simultaneously, each with his own FNT/FST pointing to the same file. Of course, the user must be validated for common files.

5. LIFT=10. Library files were discussed under the locked common files. The FNT/FST is released at return or end-of-job time, but the file space is not dropped.

6. PTFT=11. Primary terminal files are created for the terminal user only. When he issues the OLD command, he is given a copy of the IPF file with PTFT type in unlocked mode. However, since it is an IPF, any changes he makes will not affect the original copy unless a REPLACE command is issued.

   When he issues the LIB command, he is requesting either an IPF or a DPF from the User Number LIBRARY whose UI=377776B. It is equivalent to the commands GET or ATTACH, pfn/UN=LIBRARY.

   When he issues the NEW command, he gets a scratch file with FNT type PTFT in unlocked mode.

   It is important to note that the issuance of OLD, NEW or LIB will drop all files local to this TTY unless the NODROP command is issued immediately following.

7. PMFT=12. When the user issues an ATTACH command, he gets the file pointed to by an FNT of type PMFT. If the file is attached in read mode, then many users can each get an FNT of type PMFT pointing to the same file. If a user desires the file in write mode, he will have the only FNT pointing to the file. See the TEFT discussion in chapter 5 for the procedure on write mode PF attaching.

8. FAFT=13. Fast attach files are files which have an FNT always in the FNT table. PFM searches the FNTs first on an ATTACH command, and if it finds it there, PFM can save the catalog search. See PFM for more detail on FAFT files.

10.2.2 LFM

LFM is a PP routine called in the manner as PFM.

1. A CP user can use the appropriate macro which generates an RA+1 call to LFM.

2. A CP user can use the appropriate control card, which causes either RESEX or FILES to be loaded in the users FL. RESEX is loaded for the ASSIGN and REQUEST control card. See chapter 9 for the discussion of tape assignment. Local RMS assignment is the same procedure as for tape assignment. FILES will use the appropriate macro RENAME, COMMON, RELEASE, etc. to generate an RA+1 call to LFM.

When called to a CP, LFM will locate or create the FNT/FST for the desired file. It will make the appropriate changes in the FNT/FST entry for the function specified in the call. LFM also will interface to RESEX or MAGNET if necessary.

It is interesting to note that the routine FILES also does file skipping, rewinding and WRITER and WRITEF commands. It will call CIO for these tasks.

The terminal user must enter the BATCH subsystem in order to use the LFM functions.

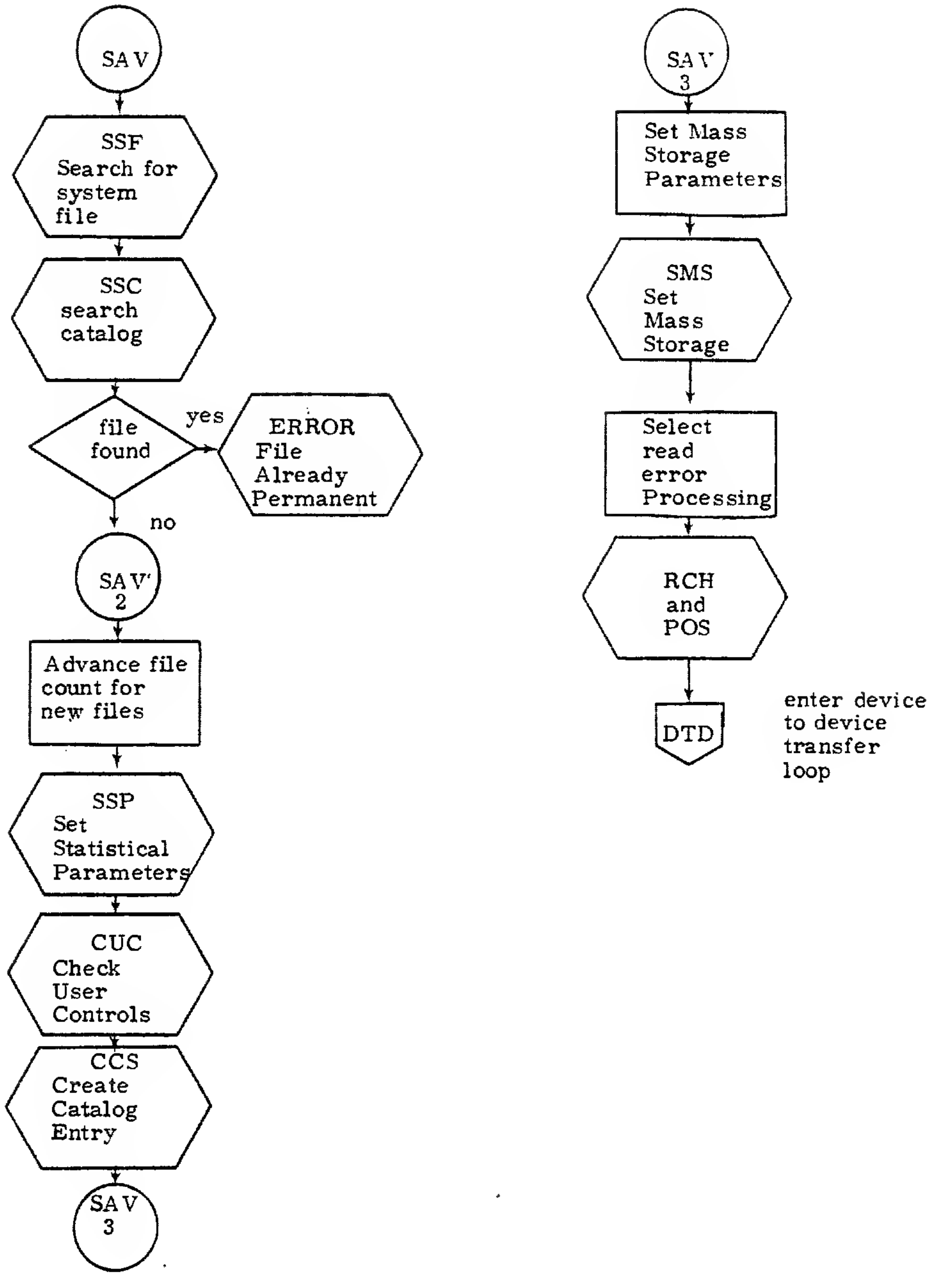


Figure 10-10. SAVE Command Processing

TABLE 10-2. LFM OVERLAYS (Cont'd)

Code	Function	Overlays	Entry Point
20	Access library file	LFM	ALF
21	Attach control statement file	3LF	ACS
22	Enter control statement file	3LF	ECS
23	Position control statement file	3LF	
24	LABEL request	LFM	LBI
25	Get all local FNTs	3LC	GTF
26	Request tape assignment	3LC	RTA
27	Enter VSN entry file	3LC	VSN

Overlay 3LA is the error processing overlay for LFM. All of the 3L(x) overlays are loaded at location OVL defined in the main LFM routine. (Currently, OVL = 1534).

Some of the macro definitions for the above functions are in SYSTEXT, while others are in COMCMAC. For instance, GETFNT is defined in COMCMAC rather than in SYSTEXT since it is only used by the CHECKPT routine. Others defined in COMCMAC include: ACCSF, ENCSF, and PSCSF. All of the macros are described in the KRONOS 2.1 Reference Manual. However, one macro, SETID, requires some additional explanation. This function causes the upper 6 bits of the FST entry for an existing file to be updated to contain the ID code specified by the macro call. The ID code is used to direct a file to a particular device identified with the same ID. For instance, a printer may be assigned an ID of 5 by the operator. This is done with the command:

LPxx, 5.

where, xx is the EST ordinal of the printer. A user may specify that his output file be disposed to that particular line printer by use of SETID.

An outline of the LFM memory map is given in Figure 10-11. The map represents LFM code prior to loading a 3L(x) overlay.

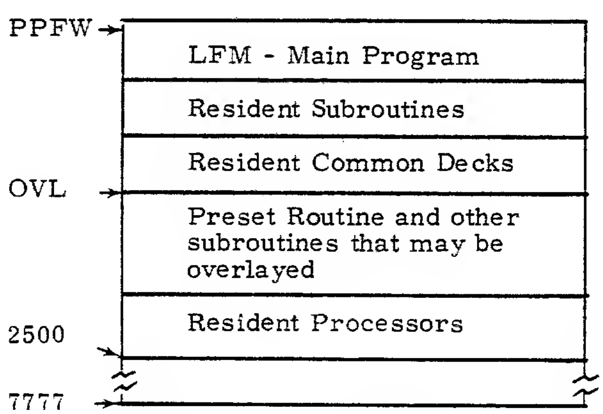


Figure 10-11. LFM Memory Map

The main program (LFM) calls the preset subroutine (PRS) and then jumps to the appropriate subroutine to process the requested function (A 3L(x) overlay is loaded, if required). LFM also contains the common return point, LFMX, from all function processors. LFMX sets the file status "not busy", completes the FET status, and drops the PP.

The resident subroutines are called from the various function processors and include the following subroutines:

- CKE - Check error processing (bit 44 of FET+1)
- DEQ - drop (release) equipment
- DRF - drop file (call 0DF)
- EFN - enter new file name in FNT (call 0BF)
- ABT - abort job
- ERR - process error (call 3LA)
- RCL - recall LFM
- SVF - Search for VSN entry file

Resident common decks include:

- COMPSAF - search for assigned file
- COMPSFB - set file busy
- COMPVFN - verify file name

The preset routine checks the input register for a valid call to LFM, determines what overlay is needed to process the requested function, initializes some memory cells, and returns to LFM (the main program). Other subroutines, besdies PRS, that are overlayed include the following:

- CRS - determine if RESEX has been called
- TFCN - function table. Specifies the entry point address of the routine to process the requested function, and in which overlay that entry point is defined.

Resident processing routines are those routines not requiring the loading of a special 3L(x) overlay. Resident in LFM due to high volume use. They include:

- ALF - access library file - function 20
- RQI - request equipment - function 14.
- AEI - assign equipment - function 15.
- LBI - LABEL request - function 24.

	18	6		12	6	18
RA+1	LFM	R		FN	ID	FET Address

The following is a dump of various tracks and FL of the PF system:

EXAMPLE OF PERMANENT FILES ORGANIZATION

Foil number	Description
1 So, the last track is 257, word 4373, byte 3. Track 260 word 4374, byte 0 is the 1st track of a Direct PF chain which continues on track 564, word 4455 byte 0 and continues on track 565 etc. This is the IAPF, indirect access PF chain. Track 261 word 4374, byte 1 is the 1st track of Permit information. Track 262 word 4374, byte 2 is the system label track.	ABSDMP of MST/TRT for SYSTEM and PF device notice that track 0, the label track points to track 237, also word 4306 says mask is 377 and bit 11 is off so catalog tracks not contiguous with label track, hence catalog tracks begin in track 237. Since this is an 844, DI-1 type, there are 20B catalog tracks. See system I/0 section 7 for foil of label track. See word 4304. 1st track of indirect PF's on track 260. Label track on track 0. 1st track of permit buffers is 261. There are 20B catalog tracks. System table track is 262.
2	CATLIST of files for user MLO.
3	DUMPTK of catalog track for user MLO, OPL, ALSON, USER, USERALL (See foil from MODVAL for user list). Also copy of MST in lower right. Note that this is a continuation of the Label track so track 237 begins in sector 0.

Description of OPL

Word 0 pfn = OPL user index = 2

1. File length = 1060000B sectors, First track = 263, first sector = 4000, indicates that this is a direct PF.
2. Random index for permission buffer = 0, no permission creation date biased by 1970. June 21, 1974.
3. The file has been accessed 36 times, modification date same as creation date.
4. CT=2 means Public file, mode = 100 means Read permission, dn (device number) = 0 means file is on Master device (i.e., this device) last access was July 3, 1974 at 14.42.20.

Description of NEW070.

This is a hole. It was a direct access file. 1st sector = 4000. Hence, when it was purged all tracks associated with this file were released back to the system.

Description of MLOPL

Word 0 pfn = MLOPL UI = 1

1. File length = 255 sectors, first track = 613, first sector is 27. This is an indirect file since bit 11 of first sector field is off. Note that each indirect PF is terminated by an eoi sector which is not included in the file length sector count (see FAST)

EXAMPLE OF PERMANENT FILES ORGANIZATION (Continued)

2. No permission file. Created June 23, 1973.

3. Accesses = 17, no modifications.

4. Last access date July 3, 1974.

   CT = 0 means private file, mode = 0 means
   Write permission, dn=0 since this is an indirect PF and must reside on this device.

Description of FAST

Word 0 pfn = FAST, UI = 1

1. Length = 1, 1st track = 574, 1st sector = 37, indirect PF.

2. Creation July 1, 1974.

3. Modified July 3, 1974, accesses = 6.

4. Last access July 3, 1974.

Description of LGO

Word 0 pfn = LGO UI = 0 this is a hole.

1. Length = 3 sectors This was an indirect PF

2. N/A

3. No. of accesses before purge was 2.

4. N/A

Note: There are 2 entries for JOKE. The first entry is a hole of 155 sectors. The second entry was created when JOKE was expanded to 3255 sectors and replaced.

There is only 1 entry for JET. JET was 355 sectors and was modified to 1 sector and replaced. The entry for JET was used for the replace and a new hole entry was created at the end for the residue.

SE = Sector number 0, 1, & 2 contain catalog entries and SE = 3 is an EOI. The System sector is Track 0, SE=0 since this is a part of the LABEL track chain.

Dayfile showing PERMIT commands and alternate CATLIST of MLO for ALSON and for OPL and CATLIST of MLO.

5. DUMPTK of catalog track for users MLO, OPL, etc.

- Description of file POSTERS

Word 2 random index is 1 see Foil 6

- Description of file SIGN

Word 2 random index is 2 see Foil 6

- Description of file FAST

Word 2 random index is 3 see Foil 6

EXAMPLE OF PERMANENT FILES ORGANIZATION (Continued)

- Description of file CMRDECK

Word 2 random index is 4 see Foil 6

- Description of file PW

Word 0 pfn = PW UI = 1

1. Length is 255, 1st track is 703, first sector is 41, it is an indirect PF.
2. Random index = 0, no permit buffers.

3, 4. Same as other descriptions.

5. Password is RT.

Note: There are all 10 word catalog buffers. Remember originally we had pfn = JET of 355 sectors. Then we replaced it with JET of 1 sector, which created a hole of 354. Now, we've filled the hole with PW and have created a new hole for the residue at the end of the catalogs in SE = 3. So old hole of 354 sectors is reallocated as PW with 255 sectors of data and 1 eoi sector, plus new hole of 76 sectors. So, 255B + 1B+ 76B = 354B.

Note that the eoi sector is now SE = 4.

DUMPTK of First track of permission buffers.

- Description of sectors

SE	Description
0	System sector
1	Random index = 1 means permit buffer for pfn = POSTERS

Word 0 Random index for linking = 0. So all permitted users are in this sector. UI = 1, who owns the file

1. Date and time
2. Permitted user number = OPL, UI = 2.
3. Accesses = 0, mode = 1 = R.
4. Permitted user number = ALSON, UI = 3.
5. Accesses = 1, Mode = 1 = R.
6. Eor

SE	Description
2	Random index = 2 means permit buffer for pfn=SIGN.

Word 0 no linking, UI = 1, who owns this file
1 date and time
2 permitted user = ALSON, UI = 3
3 accesses = 1, mode = 3 = E
4 eor

EXAMPLE OF PERMANENT FILES ORGANIZATION (Continued)

SE	Description
3	Random index = 3 means permit buffer for FAST

Word 0, 1, 2 same as above
3 mode = W
4 eor

4	Random index = 3 means permit buffer for CMRDECK

Word 0, 1, 2 same as above
3 mode = E. Note the last mode was stored.
4 eor

5	eoi

MST (1)

ABSOLUTE DUMP FROM 004300 TO 005700 PAGE 1

004300	00402234062400002324	SRIFT ST	77770153000153000153	↑↑AS AS AS	00000000000000000000	
004303	00000000000000000000		42600000476100204262	7≡5 7(P71	70420000000000030005	+7 Ċ E
004306	15172222110000400377	HURRI SC↓	00000000000000000000		00000000000000000000	
004311	00000000000000000000		00000000000000000000		00000000000000000000	
004314	00000000000000000000		00000000000000000000		00000000000000000000	
004317	00000000000000000000		42374002400340044017	745B5C5D50	40054006400740100017	5E5F5G5H 0
004322	40114012401340140017	5I5J5K5L 0	40154016401740200017	5M5N5O5P 0	40214022402340240017	5Q5R5S5T 0
004325	40254026402740300017	5U5V5W5X 0	40314032403340340017	5Y5Z5051 0	40354036403740400017	52535455 0
004330	40414042404340440017	56575859 0	40454046404740500017	5+5-5*5/ 0	40514052405340540017	5(515$5= 0
004333	40554056405740600017	5 5,5.5≡ 0	40614062406340640017	5[515↑5≠ 0	40654066406740700017	5→5∨5∧5↓ 0
004336	40714072407340740017	5↓5<5>5≤ 0	40754076407741000017	5≥5¬5↑6 0	41014102410341040017	6A6B6C6D 0
004341	41054106410741100017	6E6F6G6H 0	41114112411341140017	6I6J6K6L 0	41154116411741200017	6M6N6O6P 0
004344	41214122412341240017	6Q6R6S6T 0	41254126412741300017	6U6V6W6X 0	41314132413341340017	6Y6Z6061 0
004347	41354136413741400017	62636465 0	41414142414341440017	66676869 0	41454146414741500017	6+6-6*6/ 0
004352	41514152415341540017	6(6)6$6= 0	41554156415741600017	6 6,6.6≡ 0	41614162416341640017	6[6]6↑6≠ 0
004355	41654166416741700017	6→6∨6∧6↓ 0	41714172417341740017	6↓6<6>6≤ 0	41754176417742000017	6≥6¬617 0
004360	42014202420342040017	7A7B7C7D 0	42054206420742100017	7E7F7G7H 0	42114212421342140017	7I7J7K7L 0
004363	42154216421742200017	7M7N7O7P 0	42214222422342240017	7O7R7S7T 0	42250065422742300017	7U ∧7W7X 0
004366	42314232423300350017	7Y7Z70 2 0	00010001000142400017	A A A75 0	42414242424342440017	76777879 0
004371	42454246424742500017	7+7-7*7/ 0	42514252425342540017	7(7)7$7= 0	42544256425700000017	7 7,7. 0
004374	45640005427142700417	+≠ A7+7+≠0	00050044004000407417	E 9 5 5≤0	42720023427342740017	7< 57>7< 0
004377	42754276427743000017	7≥7¬718 0	43014303000243040017	8ABC 8AD 0	43054306430743100017	8E8F8G8H 0
004402	43114312431343140017	8I8J8K8L 0	43154316431743200017	8M8N8O8P 0	43214322432343240017	8Q8R8S8T 0
004405	43254326432743300017	8U8V8W8X 0	43314332433343340017	8Y8Z8081 0	43354336433743400017	82838485 0
004410	43414342434343450017	8687888+ 0	45714346434743500017	++8-8*8/ 0	43514352435343540017	8(8)8$8= 0
004413	43554356435743600017	8 8,8.8≡ 0	43614362436343640017	8[8]8↑8≠ 0	43654366436743700017	8→8∨8∧8↓ 0
004416	43714372437343740017	8↓8<8>8≤ 0	43754376437744000017	8≥8¬8↑9 0	44014402440344040017	9A9B9C9D 0
004421	44054406440744100017	9E9F9G9H 0	44114412441344140017	9I9J9K9L 0	44154416441744200017	9M9N9O9P 0
004424	44214422442344240017	9Q9R9S9T 0	44254426442744300017	9U9V9W9X 0	44314432443344340017	9Y9Z9091 0
004427	44303777444037770017	934↑954↑ 0	44414442444344440017	96979899 0	44454446444744500017	9+9-9*9/ 0
004432	44514452445344540017	9(9)9$9= 0	44554456445744600017	9 9,9.9≡ 0	44614462446344640017	9[9]9↑9≠ 0
004435	44654466446744700017	9→9∨9∧9↓ 0	44714472447344740017	9↓9<9>9≤ 0	44754476447745000017	9≥9¬91+ 0
004440	45014502450345040017	+A+B+C+D 0	45054506450745100017	+E+F+G+H 0	45114512451345140017	+I+J+K+L 0
004443	45154516451745200017	+M+N+O+P 0	45214522452345240017	+Q+R+S+T 0	45254526452745300017	+U+V+W+X 0
004446	45314532453345340017	+Y+Z+0+1 0	45354536453745400017	+2+3+4+5 0	45414542454345440017	+6+7+8+9 0
004451	45454546454745500017	++++-+*+/ 0	45514552455300570017	+(+)+$. 0	45700002000000000014	+↑ A L
004454	00000000000000000000		45654577000000000014	+≠+↑ L	45734572455400440017	+>+<+= 9 0
004457	00000000000046000001	- A	46014613000000000014	-A-K 1.	00000000000000000000	
004462	00000000000043440001	89 A	00000000000000000000		00000000000000000000	
004465	00000000000000000000		00000000000000000000		00000000000000000000	
004470	00000000000000000000		00000000000000000000		00000000000000000000	
004473	00000000000000000000		00000000000000000000		00000000000000000000	
004476	00000000000000000000		00000000000000000000		00000000000000000000	
004501	00000000000000000000		00000000000000000000		00000000000000000000	
004504	00000000000000000000		00000000000000000000		00000000000000000000	
004507	00000000000000000000		00000000000000000000		00000000000000000000	
004512	00000000000000000000		00000000000000000000		00000000000000000000	
004515	00000000000000000000		00000000000000000000		00000000000000000000	
004520	00000000000000000000		00000000000000000000		00000000000000000000	
004523	00000000000000000000		00000000000000000000		00000000000000000000	
004526	00000000000000000000		00000000000000000000		00000000000000000000	
004531	00000000000000000000		00000000000000000000		00000000000000000000	
004534	00000000000000000000		00000000000000000000		00000000000000000000	
004537	00000000000000000000		00000000000000000000		00000000000000000000	
004542	00000000000000000000		00000000000000000000		00000000000000000000	
004545	00000000000000000000		00000000000000000000		00000000000000000000	
004550	00000000000000000000		00000000000000000000		00000000000000000000	
004553	00000000000000000000		00000000000000000000		00000000000000000000	

TRT

Some other Direct PF Files

end of Label/Catalog track

$$237 = \underbrace{010\ 011}_{\text{word } 47}\ \underbrace{1\ 1}_{\text{byte } 3}$$

TRT = 4320
+ 47
4367

Label/Catalog track begins at 4367 byte 3 and ends at 4373 byte 3

```
CATALOG OF                      74/07/03. 12.38.33.  PAGE   1
 FILE NAME(S)

BOBSPL    POSTERS   HLOPL     SIGN      CHRDECK   FAST      IPRDECK
DUMPTK    TEST      JOKE      JET

      11 FILE(S)
```

(2)

UI=2

1st sector = 4000
means Direct PF

EXAMPLE of CATALOG TRACK

3

DUMPTK (TK=237)   DUMPTK - VER. 1   74/07/03. 12.39.07.   PAGE 1

WORD	TK=237 SF=0 B1=1	B2=100	TK=237 SF=1 B1=2	B2=100	TK=237 SF=2 B1=3	B2=100
0	17201400000000000002	OPL B	01022304152000000000	ABSDMP	14071700000000000000	LOD
1	01060000000004263000	AF 7;5	00000020000045540000	P •=	00000003000045737777	C •>!!
2	00000000050621105772	DFDH.<	00000000030625111245	CFUIJ•	00000000040702132721	DGBKWD
3	00000034040621105772	JDFDH.<	00000001030625111245	ACFUIJ•	00000002040702132721	BDGBKWQ
4	02010000040703144220	BA DGCL7P	00000000030701136471	CGAK#•	00000000040702142310	DGBLSH
10	16052733423300000000	NEWU70	24142000000000000000	TLP	11202204050313000001	7PRDECK A
11	00000000000045574000	•.5	00000002000045540021	B •= Q	00000002000045730041	B •> 6
12	00000000030702150550	COHME/	00000000030623104622	CFSHVR	00000000040702142335	DGBLS2
13	00000000030702150550	CGBME/	00000004030623104622	DCFSHVR	00000004030702146555	DCGBL•
14	00000000030702150550	CGBME/	00000000030701132746	CGAKW-	00000000030702146663	CGBLV!
20	02170222201400000001	BGBSPL A	22050704152000000000	REGDMP	04251520241300000001	DUMPTK A
21	00000030700004260002	CW 7≡ B	00000005000045540024	E •= 7	00000503000045730045	FC •> •
22	00000000030623075064	CF5G/#	00000000030623102716	CFSHWN	00000000030623102417	CFSHTD
23	00000014030623075064	NCF5G/#	00000002030623102716	BCFSHWN	00000005040703144273	EDGCL7>
24	00000000040702140223	DGBLB5	00000000030625105146	CFUH(-	00000000040703144273	DGCL7>
30	20172324052223000001	POSTERS A	22230500000000000000	R5E	24052324000000000001	TEST A
31	00000714000005540137	GN •#A4	00000002000045540032	B •= Z	00003072000045640050	X< •V /
32	00000000030623075071	CF5G/•	00000000030623104247	CFSH7•	00000000040703144346	DGCL8-
33	00000002030623075071	BCF50/•	00000005030623104247	ECFSH7•	00000000040703144346	DGCL8-
34	00000000040702131101	DGBK1A	00000000040701145155	DGAL(	00000000040703144346	DGCL8-
40	15141720140000000001	MLGPL A	03152204050313000001	CHRDECK A	12171305000000000000	JOKE
41	00000025000004130027	B •K W	00000001000045540035	A •= 2	00000155000046330036	A •0 3
42	00000000030623077213	CF5G<K	00000000030623105205	CFSHIE	00000000040703144430	DGCL9X
43	00000017030623077213	UCF50<K	00000000030623105205	CFSHIE	00000000040703144430	DGCL9X
44	00000000040703142755	UGCLW	00000000030623105205	CFSHIE	00000000040703144430	DGCL9X
50	24232347330000000000	TS570	24052324000000000000	TEST	12171305000000000001	JOKE A
51	00000102000063440132	A0 89AZ	00000000000052554000	I 5	00003255000046530041	7 •$ 6
52	00000000030623100060	CFSH ≡	00000000040702141452	DGBLL)	00000000040703144430	DGCL9X
53	00000004030621100060	UCFSH ≡	00000000040702141452	DGBLL)	00000000040703144507	DGCL+G
54	00000000030701132420	CGAKTP	00000000040702141452	DGBLL)	00000000040703144507	DGCL+G
60	23110714000000000001	SIGN A	04230000000000000000	D5	12052400000000000001	JET A
61	00000054000045720062	= •< 1	00000320000045547777	CP •=!!	00000001000047030037	A •C A
62	00000000030623102045	CFSHP•	00000000030625105316	CFUHSN	00000000040703144610	DGCL-H
63	00000002030623102045	BCFSHP•	00000001030625105316	ACFUHSN	00000000040703144634	DGCL-1
64	00000000030623103536	CFSH23	00000000030701141703	CGALDC	00000000040703144634	DGCL-1
70	04251520241300000000	DUMPTK	06012324000000000001	FAST A	00000000000000000000	
71	00000013000045720137	K •<A4	00000001000045730037	A •> 4	00000354000047030041	C• •C 6
72	00000000030623102417	CF5H7D	00000000040701146424	DGAL#7	00000000000000000000	
73	00000005030623102417	ECF5HTO	00000006040703143440	FDGCL15	00000000000000000000	
74	00000000030701134161	CGAK6[	00000000040703143540	DGCL25	00000000000000000000	

WORD	TK=237 SF=3 B1=0	B2=0
0	01001465300610752100	A L*KFH≥0
1	33206010300612033401	OP≡MXFJC1A
2	50010010036330010100	/A MCIXAA
3	36533403020006160603	3$1CB FNFC
4	02001333500300013401	H KU/C A14
5	40030505340150030001	5CEE1A/C A
6	04573407100606531071	D,1UHFF$H•
7	34061400340702000606	1FL 1GB FF
10	40030403500300013401	5CDC/C A1A
11	01001503306302001504	A MCX1B MD
12	40635163000104040200	5:((ADDB
13	17030764020017214063	OCGVB DQS!
14	51630001041637300604	((ADN4XFD
15	36303727073102001777	3X4#GYB D!

EOI sector

MST ADDRESS = 4300

MST

00363532062400002216
77770153000153000153
00000000000000000000
00000000000000000000
42604000426100204262
70420000000000040005
15172222110000400377
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000

97404700C

(4)

```
DIS044F.   74/07/03. MORRIES PERSONAL KRONOS 2.1.

14.50.04.DIS.
14.50.04.MODE(0)
14.50.05.RETURN(INPUT)
14.50.05.DIS.
14.50.12.ACCOUNT,MLO..
14.50.15.CATLIST.
14.50.15. CATLIST COMPLETE.
14.51.04.PERMIT(POSTERS,OPL=R)
14.51.17.PERMIT(POSTERS,ALSON=R)
14.51.37.PERMIT(SIGN,ALSON=W)
14.51.50.PERMIT(FAST,ALSON=W)
14.52.05.PERMIT(SIGN,ALSON=E)
14.52.47.PERMIT(CMPDECK,ALSON=E)
14.52.49.CP         0.016 SEC.
14.52.49.MS         0.005 KPR.
14.52.54.LP20       0.054 KLN.
```

changes the permission

```
CATALOG OF ALSON                74/07/03. 15.06.35.  PAGE   1
        ALTERNATE CATALOG   MLO
 FILE NAME(S)

POSTERS   SIGN      CMPDECK   FAST

       4 FILE(S)
```

```
CATALOG OF OPL                  74/07/03. 15.07.29.  PAGE   1
        ALTERNATE CATALOG   MLO
   FILE NAME ACCESS FILE-TYPE LENGTH ON CREATION LAST ACCESS LAST MOD
             MD/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

   1 POSTERS  IND. PRIVATE   295680     73/06/19. 74/07/03. 73/06/19.
                 5           READ       07.40.57. 14.53.45. 07.40.57.
```

```
CATALOG OF MLO                  74/07/03. 15.09.19.  PAGE   1

   FILE NAME ACCESS FILE-TYPE LENGTH ON CREATION LAST ACCESS LAST MOD
    PASSWORD MD/CNT INDEX  PERM. SUBSYS DATE/TIME DATE/TIME DATE/TIME

   1 AORSPL   IND. PRIVATE   127360     73/06/19. 74/07/02. 73/06/19.
                14           WRITE      07.40.52. 12.02.19. 07.40.52.

   2 POSTERS  IND. PRIVATE   295680     73/06/19. 74/07/03. 73/06/19.
                 5           WRITE      07.40.57. 14.53.45. 07.40.57.

   3 MLOPL    IND. PRIVATE   110720     73/06/19. 74/07/03. 73/06/19.
                16           WRITE      07.58.11. 15.08.00. 07.58.11.

   4 SIGN     IND. PRIVATE    28160     73/06/19. 74/07/03. 73/06/19.
                 5           WRITE      08.16.37. 14.53.20. 08.16.37.

   5 CMPDECK  IND. PRIVATE      640     73/06/19. 74/07/03. 73/06/19.
                 1           WRITE      08.42.05. 14.52.47. 08.42.05.

   6 FAST     IND. PRIVATE      640     74/07/01. 74/07/03. 74/07/03.
                 8           WRITE      12.52.20. 14.51.50. 12.28.32.

   7 TPRDECK  IND. PRIVATE     1280     74/07/02. 73/07/02. 73/07/02.
                 4           WRITE      12.19.29. 12.54.51. 12.53.45.

   8 PW       IND. PRIVATE   110720     74/07/03. 74/07/03. 74/07/03.
     RT          0           READ       15.08.54. 15.08.54. 15.08.54.
```

Character Count

1 sector = 640 words @ 10 characters.
= 640 characters

CATALOG TRK for User number 1 (5)

DUMPTK(TK=237. DUMPTK - VER. 1 74/07/03. 15.09.33. PAGE 1

WORD	TK=237 SF=0 B1=1	B2=100	TK=237 SE=1 B1=2	B2=100	TK=237 SE=2 B1=3	B2=100
0	17201400000000000007	OPL B	01022304152000000000	ABSDMP	14071700000000000000	LGO
1	01068000000042634000	AF 7 5	00000020000045540000	P +=	00000003000045737777	C +>!!
2	00000000046621105772	DFQH,<	00000000030625111245	CFUIJ+	00000000040702132721	DGBKWQ
3	00000410406621105772	6UFQH,<	00000001030625111245	ACFUIJ+	00000002040702132721	RDGBKWQ
4	02010000040432000424	BA UUZ DT	00000000030701136471	CGAK#+	00000000040702142310	DDALSH
10	16052733423300000000	NEWU70	24142000000000000000	TLP	11202204050313000001	IPADECK A
11	00000000000005574000	+.5	00000020000045540021	B += Q	00000002000045730041	A +> 6
12	00000000030702150550	CDBME/	00000000030623102622	CFSHVR	00000000040702142336	DGBLS2
13	00000000030702150550	CGBME/	00000004030623102622	DCFSHVR	00000004030702146555	DCDAL*
14	00000000030702150550	CDBHE/	00000000030701132746	CGAKW-	00000000030702146663	CGBLV!
20	02170223201400000001	BDBSPL A	22050704152000000000	REGDMP	04251520241300000000	DUMPTK
21	00000307000042600002	CG 7≡ B	00000050000045540024	E += T	00000155000045730045	A +> +
22	00000000030623075064	CFSG/*	00000000030623102716	CFSMWN	00000000030623102417	CFSHTD
23	00000016030623075064	NCFSG/*	00000002030623102716	RCFSMWN	00000005040703144273	EDGCL7>
24	00000000040702140273	DGBLBS	00000000030625105146	CFUH(-	00000000040703144273	DDCL7>
30	20172324052223000001	PDSTERS A	22230500000000000000	RSE	24052324000000000000	TEST
31	00000716000045440137	GN +*A4	00000020000045540032	R += Z	00000140000045640050	AS +V /
32	00000001030623075071	ACFSG/+	00000000030623104247	CFSH7*	00000000040703144346	DDCL8-
33	00000005030623075071	ECFSG/+	00000005030623104247	ECFSH7*	00000000040703144346	DGCL8-
34	00000000040703166555	DDCN*	00000000040701145155	DDAL(	00000000040703144346	DDCL8-
40	15141720140000000001	MLOPL A	03152204050313000001	CMRDECK A	12171305000000000000	JOKE
41	00000255000046130027	B -K W	00000001000045540035	A += 2	00000155000046330036	A -0 3
42	00000000030623077213	CFSD<K	00000004030623105205	DCFSHIE	00000000040703144430	DDCL9X
43	00000020030623077213	PCFSD<K	00000001030623105205	ACFSHIE	00000000040703144430	DDCL9X
44	00000000040703171000	DGCOM	00000000040703166457	DGCN#.	00000000040703144430	DDCL9X
50	24232342330000000000	TS5/0	24052324000000000000	TEST	12171305000000000000	JOKF
51	00000102000043440132	AB A9AZ	00000000000052554000	) 5	00000150000046530041	A/ -$ 6
52	00000000030623100060	CFSH ≡	00000000040702141452	DDBLL)	00000000040703144430	DDCL9X
53	00000004030623100060	DCFSH ≡	00000000040702141452	DGBLL)	00000000040703144507	DDCL+G
54	00000000030701132420	CGAKTP	00000000040702141452	DDBLLI	00000000040703144507	DDCL+G
60	23110714000000000001	SIGN A	04230000000000000000	DS	12052400000000000000	JET
61	00000054000045720062	= +< 1	00000320000045547777	CP +=!!	00000001000047030037	A =C 4
62	00000007030623102045	GCFSHP+	00000000030625105316	CFUH$N	00000000040703144610	DGCL=H
63	00000005030623102045	ECFSHP+	00000001030625105316	ACFUH$N	00000000040703144634	DDCL=)
64	00000000040703166524	DDCN*T	00000000030701141703	CGALDC	00000000040703144634	DDCL=)
70	04251520241300000000	DUMPTK	06012324000000000001	FAST A	20270000000000000001	PW A
71	00000113000045720137	K +<A4	00000001000045730037	A +> 4	00000255000047030041	R =C 6
72	00000000030623102417	CFSHTD	00000003040701146424	CDGAL#T	00000000040703171066	DDCDHV
73	00000005030623102417	ECFSHTD	00000010040703143440	HDGCL)5	00000000040703171066	DGCDHV
74	00000000030701134161	CGAK6(	00000000040703166362	DGCN!!	00010000040703171066	A DDCDHV
76	00000000000000000000		00000000000000000000		22240000000000000000	RT

MST

WORD	TK=237 SF=3 B1=4	B2=10	TK=237 SE=4 B1=0	B2=0 EOT sector	00401205062400002317
0	00000000000000000000		01003445300410752100	A]+XFH≥Q	77770153000153000153
1	00000074000046500144	¬ -/A9	33206010300612033401	OP≡HXFJC1A	00000000000000000000
2	00000000000000000000		50010010036330010100	/A HC1XAA	00000000000000000000
3	00000000000000000000		36533403020006160603	3$1CB FNFC	42604000476100204262
4	00000000000000000000		02001333500300013401	B KO/C A1A	70420000000000030005
5	00000000000000000000		40030505340150030001	5CEE1A/C A	15172222110000400377
6	00000000000000000000		04573407100606531071	D.1GHFF$H+	00000000000000000000
7	00000000000000000000		34061400340702000606	1FL 1DB FF	00000000000000000000
10	40030403500300013401	5CDC/C A1A	40030403500300013401	5CDC/C A1A	00000000000000000000
11	01001503306302001504	A HCX1B HD	01001503306302001504	A MCX1B HD	00000000000000000000
12	40635161000104040200	51(1 ADDB	40635163000104040200	51(1 ADDB	00000000000000000000
13	17030766020017214063	UCGVB DOS!	17030766020017214063	DCGVB DOS!	00000000000000000000
14	51630001041637300604	(1 ADN4XFD	91630001041637300604	(1 ADN4XFD	00000000000000000000
					00000000000000000000
					00000000000000000000

MST ADDRESS = 4300

97404700C

10-29

10-30

PERMIT BUFFER TRACK

(6)

```
DUMPTK(TK=261)                                  DUMPTK - VER. 1     74/07/03. 14.54.41.   PAGE - 1

WORD   TK=261  SE=0     B1=1777 B2=77       TK=261  SE=1      B1=2   B2=6                 TK=261  SE=2      B1=3   B2=4
  0    00000000000000000000                 00000000000000010000          A              00000000000000010000          A
  1    00004261000000004261   7(     7(     00000000040703166321   DGCN:Q               00000000040703166605     DGCN*E
  2    00000000040621104465   DFOH9*        17201400000000000002   DPL    B             01142317160000000003   ALSON   C
  3    00000000000000000000                 00000001040703166304   ADGCN:D              00000103040703166574   ACDGCN*T
  4    00000000000000000000                 01142317160000000003   ALSON  C             00000000000000000000              eor
  5    00000000000000000000                 00000101040703166555   AADGCN*   eor        00000000000000000000
 55    00000000000000000000                 00000100354424002036   A 29T P3             00000100354424002036   A 29T P3
 56    00000000000000000000                 20110200053320001144   PIB EOP 19           20110200053320001144   PIB EOP 19
 57    00000000000000000000                 02007004060414260100   B +DFDLVA            02007004060414260100   B +DFDLVA
 60    00000000000000000000                 13105400116610031604   KH= IVHCND           13105400116610031604   KH= IVHCND
 61    00000000000000000000                 61701156160260105000   (+1,NB=H/            61701156160260105000   (+1,NB=H/
 62    00000000000000000000                 11440415301004133052   19DMXHDKX)           11440415301004133052   19DMXHDKX)
 63    00000000000000000000                 11120504540052610305   1JED= )1CE           11120504540052610305   1JED= )1CE
 64    00000000000000000000                 20002400540037040200   P T = 4DH            20002400540037040200   P T = 4DB
 65    00000000000000000000                 46570543300534605400   -,EBXE1=*            46570543300534605400   -,EBXE1=*
 66    00000000000000000000                 11773006540012160200   1(XF= JNB            11773006540012160200   1(XF= JNB
 67    00000000000000000000                 05475600057630141003   E*, E-XLHC           05475600057630141003   E*, E-XLHC
 70    00000000000000000000                 16203272540046065400   NPZ<= -F=            16203272540046065400   NPZ<= -F=
 71    00000000000000000000                 14725400453210632300   L<= +ZHIS            14725400453210632300   L<= +ZHIS
 72    00000000000000000000                 21005400460554001471   Q = -E= L+           21005400460554001471   Q = -F= L+
 73    00000000000000000000                 54004531140034073447   = +YL 1Q1*           54004531140034073447   = +YL 1Q1*
 74    00000000000000000000                 34453446346134626063   1+1-1(1)=1           34453446346134626063   1+1-1(1)=1
 75    00000000000000000000                 60312000627334332000   =YP )>10P            60312000627334332000   =YP )>10P
 76    00000000000000000000                 55663434300402000437   V))XDB D4            55663434300402000437   V))XDB 04
 77    00000000000000000000                 02000606010061320100   B FFA (ZA            02000606010061320100   B FFA (ZA

WORD   TK=261  SE=3     B1=4    B2=4        TK=261  SE=4      B1=5   B2=4                 TK=261  SE=5      B1=0   B2=0      EOI
  0    00000000000000010000          A      00000000000000010000          A              01001465300610752144   A L-XFHZQ
  1    00000000040703166362    DGCN:1       00000000040703166457     DGCN*.              33206010300612073401   0P=HXFJC1A
  2    01142317160000000003  ALSON   C      01142317160000000003   ALSON   C             50010010036370010100   /A HC1XAA
  3    00000000040703166362    DGCN:1  eor  00000003040703166457     CUGCN*.             36573403020006140603   3$1CB FNFC
  4    00000000000000000000                 00000000000000000000                         02001333500300013401   B K0/C A1A
  5    00000000000000000000                 00000000000000000000         eor             40030505340150030001   5CFE1A/C A
  6    00000000000000000000                 00000000000000000000                         04573407100606531071   D,1GHFF$H+
  7    00000000000000000000                 00000000000000000000                         34061400340702000606   1FL 1GB FF
 10    00000000000000000000                 00000000000000000000                         40030403500300013401   5CDC/C A1A
 11    00000000000000000000                 00000000000000000000                         01001503306302001504   A MCX1A HD
 12    00000000000000000000                 00000000000000000000                         40635163000104040200   5((( ADOB
 13    00000000000000000000                 00000000000000000000                         17030766020017214063   OCDVB 0Q5)
 14    00000000000000000000                 00000000000000000000                         51630001041637300604   (( ADN4XFD
 15    00000000000000000000                 00000000000000000000                         36303727073102001777   3X4WGYA O(
 16    00000000000000000000                 00000000000000000000                         02001703076202001771   B DC01A DO
 17    00000000000000000000                 00000000000000000000                         03433027313004233052   CAXWYXDSX)
 20    00000000000000000000                 00000000000000000000                         11020413140044633730   1BDKL 914X
 21    00000000000000000000                 00000000000000000000                         06043630372707040200   FD3X4WGDB
 22    00000000000000000000                 00000000000000000000                         17770370020014531462   D1C+B L$L1
 23    00000000000000000000                 00000000000000000000                         01001267020014533004   A JAB L$X0
 24    00000000000000000000                 00000000000000000000                         02000446305211020514   B D-X)1BEL
 25    00000000000000000000                 00000000000000000000                         30053411300622003777   XF11XFA 41
 26    00000000000000000000                 00000000000000000000                         34173007341314430200   1JXO1KL0B
 27    00000000000000000000                 00000000000000000000                         03643057040262202000   C#X.DB1PP
 30    00000000000000000000                 00000000000000000000                         00000404143401001301     DDL1A KA
 31    00000000000000000000                 00000000000000000000                         01001374500005751006   A K-/_E2HF
 32    00000000000000000000                 00000000000000000000                         06071400546300014463   FGL =1 A91
 33    00000000000000000000                 00000000000000000000                         01001547560016550100   A M*, N A
 34    00000000000000000000                 00000000000000000000                         13320100170240630406   K7A DB510F
```

97404700C

11.0 INTRODUCTION

The reader must be familiar with the ACCOUNT and CHARGE cards, and the information on MODVAL and PROFILE in Part IV, Section 1 of the Installation Handbook.

Validation files are used to validate users on the system. Validation defines and controls the following:

1) Who can use the system
2) What they can use (hardware and software)
3) To what extent they can use it.

Every user of the system (if VALIDATION is enabled) must have a valid account number. From a batch environment, the second card must be an ACCOUNT card. This card causes the routine ACCFAM to be loaded. (See Section 5, number 7, "VAL=" special entry points). ACCFAM will access the VALIDUX file and use 0AV via CPM to verify this account number. If valid, ACCFAM will set up the validation information into the Control Point Area (CPA) and enter this job into the system.

If the CCNR (bit 7) of the access word is not set, the user must be further validated by the CHARGE routine. In this case, the third card must be a CHARGE card. This card causes the routine CHARGE to be loaded ("VAL=" SEP). CHARGE will access the PROFILO file and verify the CHARGE card for charge number and project number. If the user is validated, the job is entered into the system.

The validation procedure allows the system to do the following.

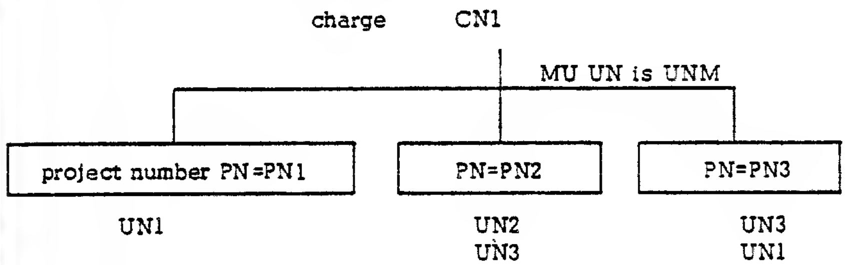
1. Determine if a user should be allowed to use the system.
2. By use of the ACCOUNT dayfile, the user can be charged for his use of the system.
3. Each user can be validated for only certain resources, i.e., restricted in his use of the system.
4. By mapping each user onto a specific user index UI, the system can maintain PFs for each user, and easily control access and absolute security for each user's PFs.

Thus, each user is given a user number UN, which is a unique seven-character name. The UN, when validated, will map the user to a specific UI (usually unique, but may be duplicated by the staff via the FUI command at MODVAL time). Each UI, then, will map to a specific set of PFs. This info is kept on the VALIDUX file which can be changed only by staff personnel at the system console.

The user may be further restricted by the use of the CHARGE system.

In this case, one user number becomes a MASTER USER (MU). The staff builds a skeleton PROFILO file, and the master user can access the file directly via BATCH or TTY and modify his charge system operation.

For example, if the MU for charge number CN1 is UNM and he has UN1, UN2, and UN3 working for him, he can specify 3 projects as follows:

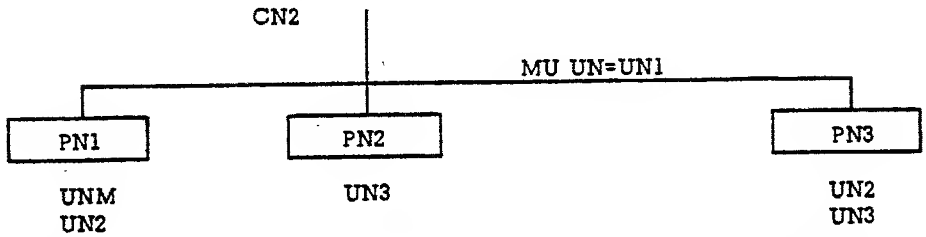


This means:

1. UN1 can use PN1 or PN3 and is the only user, other than the MU, who can use PN1.
2. UN2 can only use PN2.
3. UN3 can use PN2 or PN3.
4. UNM, the MU can use PN1, PN2 or PN3.
5. All the charges will be sent to UNM who can get a breakdown of the use on these projects. These projects can be restricted as to time of day they can be used, etc.

Each UN, by the way, will have his own PFs if each UN has a unique UI. If they all had the same UI, then they could all use the same PFs. This MU is not to be confused with the PF master user (see Chapter 10). If MU UN was UN* instead of UNM, this MU would also be a

PF master user of UN1, UN2, and UN3. To take this example even further, if UN1 was the MU for charge CN2 and his workers were UN2, UN3 and UNM (the MU for CN1), then UN1 could:



Note that in this case the project numbers are the same characters as before, but since they are under CN2, they are different than those under CN1.

This means:

1. UNM and UN2 can use PN1
2. UN3 is exclusive user (except for the MU) of PN2.
3. UN2 and UN3 can use PN3.
4. MU UN1 can use PN1, PN2, or PN3.

Now, UNM is MU for CN1 but is a controlled user for CN2. UN1 is MU for CN2, but is a controlled user for CN1.

This whole procedure allows a project to be broken into parts and to have the charges disbursed correctly.

For example, if a team of programmers with a project leader is coding a system consisting of 3 logical parts - say INPUT, PROCESSING, and OUTPUT - the project leader could get a charge number from the computer staff. He could then build a charging system similar to the example given above. Then he could tell each programmer which project number to use when on the system. At the completion of the project, the charges would be conveniently grouped into the three logical parts: INPUT, PROCESSING, and OUTPUT.

The following discussion highlights the procedure of validating users.

As has been shown in Figure 6-8, 1AJ, at Begin Job processing, will set the VAL= bit in the UIDW of the CPA if:

1. SSTL in CMR indicates ACCOUNT enabled. (See installation Handbook, Section 5.2. Part II for table of ACCOUNT/VALIDATION enabling/disabling).

2. Origin is BCOT or EIOT.

Then, IAJ loads UIDW as follows:

```
          12           12          12                       12
UIDW = | 0------0 | 0------0 | 0------0 |   0040   | 0----------0 |

              byte 3 is     0040  =  000 000 100 000
                                     222 211 111 111
                         bit count   321 098 765 432
```

which is:

```
                                        1  1
                                        7  6                  0
UIDW= |          UN=0                  |  |       UI=0         |
                                         |
                                   VAL= bit 17 = ON
```

Now, whenever this job is advanced, IAJ will see that the VAL= bit is on, and will abort the CP, unless the control card specifies VAL = SEP.

For the flow of Validation see Appendix A

Step #

A. Program with a Val = SEP.

B. Job Flow, I. Scheduler, 1 start-up.

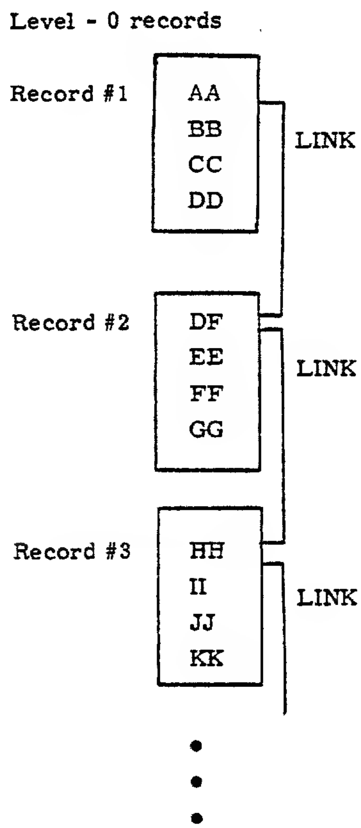
C. VALIDATION

11.1 VALIDUX AND PROFILO FILES

VALIDUX and PROFILO are Tree File Structures (TFS) and are very similar. The exact format of each file is given later. These files are created by the action of MODVAL or PROFILE and are written to a file. The use of random address allows similar levels to be linked, and allows a higher level to point to a lower level on the file.

All account and charge numbers are stored on the file in alphabetical order so that a direct search of the file can be performed.

The direct search of the files uses the following method. (Figure 11-1). As each Level - 0 Block is read, the Last Account Number (LAN) is compared with the Account Number (AC) for which this search is being made. If the LAN is less than the AC, the next level - 0 block is read. In this manner, no more than one word of each Level - 0 PRU is read if the AC is not on it. When the LAN is greater than or equal to the AC, the AC is compared with each AC from the file in backward order. When the AC from the file is again less than the AC for which the search is being made, there is a match.



Direct Access Files

If one wishes to find the Level - 0 entry for the account number IM, he must:

1) Read record 1 compare
   DD < IM

2) Read record 2 compare
   GG < IM

3) Read record 3 compare
   KK > IM

4) Move up the entries in record 3
   a) JJ > IM
   b) II < IM

The random address (RA) of this entry will point to the Level - 1 record with account number IM.

Figure 11-1. Example of Search

Once the Level - 0 match is found, no more than one Level - 1 record and one Level - 2 record will ever be necessary. (In the case of PROFILO also one Level - 3 record will need be read).

The records are 1 PRU in length and each level is linked to all records of the same level (except VALIDUX where all Level - 2 records are complete within themselves).

Level - 1 records are not normally linked. Level - 0 records are only created during a create or restructure run; during updates, changes are made to Level - 1 and Level - 2 records only. When too many account numbers are added to one Level - 1 (Level - 0 points to an overflowing Level - 1), the information overflows to a new Level - 1 record. In this case, the old Level - 1 record is linked to the new Level - 1 record. When this happens, MODVAL will issue a message to this effect to the dayfile. The user should then restructure the file (OP=R).

11.2 DEFINITIONS

The following terms are defined to enable a better understanding of their use in this section. They are not code defined.

11.2.1 Random Address (RA)

An RA on a Mass Storage (MS) device is the relative PRU number from the start of the file. Sector 0 on the first track is the system sector; therefore, zero is never a valid RA. Table 11-1 illustrates the relationship between RA and the actual disk addresses when writing a file consisting of 15 sectors (1 sector equals 1 PRU) using an MS with eight sectors-per-track.

TABLE 11-1. RELATIONSHIP BETWEEN RA AND THE ACTUAL ADDRESSES

Track Number	Sector Number	RA	Description
1	0	--	System
1	1	1	Data
1	2	2	Data
1	3	3	Data
1	4	4	Data
1	5	5	Data
1	6	6	Data
1	7	7	Data
2	0	10	Data
2	1	11	Data
2	2	12	Data
2	3	13	Data
2	4	14	Data
2	5	15*	EOR
2	6	--	Open
2	7	--	Open

* Note that RA 15 is the EOR sector

The common deck COMPCRA contains the routine CRA which will convert an RA to a track and sector address.

11.2.2 Linking Words (LW)

Equal level number blocks can be linked together using the RA. If there are n Level - 0 records, each Level - 0 record can link to its successive Level - 0 record. The last Level - 0 record has the linking byte = 0.

If, in the previous example, there were 2 Level - 0 blocks (blocks are synonomous with records), 4 Level - 1 blocks, and 7 Level - 2 blocks, the relationship of RA, actual address, and links would be as illustrated in Table 11-2.

TABLE 11-2. RELATIONSHIP OF RA - ACTUAL ADDRESS - LINKS

Track Number	Sector	RA	Level	Link RA	Description
1	0	--	--	--	System
1	1	1	0	2	Data
1	2	2	0	0	Data
1	3	3	1	0	Data
1	4	4	1	0	Data
1	5	5	1	0	Data
1	6	6	1	0	Data
1	7	7	2	0	Data
2	0	10	2	0	Data
2	1	11	2	0	Data
2	2	12	2	0	Data
2	3	13	2	0	Data
2	4	14	2	0	Data
2	5	15	2	0	EOR
2	6	--	--	--	
2	7	--	--	--	

NOTE

On VALIDUX, Level - 2 records cannot be linked.
Level - 1 records are linked only on overflow.

On PROFILO, file Level - 3, which corresponds to VALIDUX Level - 2, can be linked on overflow.

Table 11-3 illustrates what happens if someone updates the file and in the process needs to add two Level - 1 (from overflow) and four Level - 2 blocks.

TABLE 11-3. RESULTS OF UPDATE

Track Number	Sector	RA	Level	Link RA	Description
1	0	--	-	--	System
1	1	1	0	2	Data
1	2	2	0	0	Data
1	3	3	1	0	Data
1	4	4	1	16	Data
1	5	5	1	0	Data
1	6	6	1	17	Data
1	7	7	2	0	Data
2	0	10	2	0	Data
2	1	11	2	0	Data
2	2	12	2	0	Data
2	3	13	2	0	Data
2	4	14	2	0	Data
2	5	15	2	0	Data
2	6	16	1	0	Data
2	7	17	1	0	Data
3	0	20	2	0	Data
3	1	21	2	0	Data
3	2	22	2	0	Data
3	3	23	2	0	Data
3	4	24	-	--	EOR
3	5				

If a user number is removed from the file, the file is not restructured unless a MODVAL(OP=R) is used. In this case, the Level - 1 entry is eliminated and those below move up and the Header changes; Level - 0 and Level - 2 remain unchanged. (VALIDUX flags empty entries by the VALINDX file). Alphabetic order is always guaranteed even if a shuffling of data on the PRUs is required.

11.2.3 AC is the account number for which the search is being made. (i.e., the account number from ACCOUNT card).

11.2.4 DATA is the account number at the present position of the Validation file.

11.2.5 UI is the user index

11.2.6 CM is the central memory (60-bit words).

11.2.7 PF is the permanent files.

11.3 MODVAL (VALIDUX AND VALINDX FILES)

MODVAL provides creation, modification, and displays for the system Validation file. This file contains basic information necessary to validate a user's access to the system, and provides some controls on system resource usage. More comprehensive monitoring of resources can be achieved by using an additional "USER PROFILE" Validation file. For details refer to USER PROFILE CONTROL, Part IV, Section I of the Installation Handbook.

MODVAL is a system utility which is used to create and maintain the special system files VALIDUX and VALINDX. VALIDUX is a direct access file, and VALINDX is a direct access permanent file which resides under user index 377777B. VALINDX contains a record of which user indices have been assigned, while VALIDUX contains user validation information, which - when referenced through an ACCOUNT card - will define the users permanent file index and system access permissions. The VALIDUX file is a tree-structured file indexed by account numbers into a two level structure. A VALIDUX file is required for permanent file usage and one must exist for each family in the system. The control card call is:

```
MODVAL(PI-F1,...,PN=FN,...,PN)
```

MODVAL has two other entry points which are accessed by the two controls cards:

```
PASSWOR (OLDPW, NEWPW)
LIMITS.
```

The control cards are described in the KRONOS 2.1 Reference Manual.

Specific notes on options available are:

1. For the C, U, R and S options, the user must supply his own files (VALIDUX, VALINDX), unless the user is validate to use the Force fast attach status (FA) parameter (OP=U, R and S).
2. Under the K, Z, 1 and L options, the system validation file is manipulated. K, Z and L require (SYOT) validation.
3. When a new "VALIDUX" file is created under UI 377777B, the "1SF" program must be run to allow the system access to the file.
4. To attach the "VALIDUX" file to a control point, 1SF (R=VALIDUX) must be run to release "VALIDUX" from fast attach status.

MODVAL will create or update the VALIDUX file either by reading a file of input data or by accepting commands directly from the operators console via the K display (See Section 18, K-Display Programming).

If a user number is deleted from the system and the UI is returned to the available UIs, all the permanent files associated with that UI are not automatically purged.

These permanent files will become available to a new user who is assigned to this UI. The permanent files are also available to a DIS job, which uses the SUI command (see Operators Guide). Normally, new users are assigned UIs sequentially so UI holes would not be assigned unless this specific UI is specified. If a user number is going to be deleted from the system, it is wise that the analyst also uses the PURGALL command on this UI.

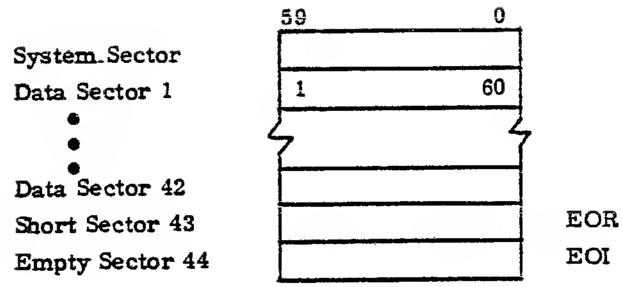
The VALINDX bits once set, remain set even if the corresponding UI is deleted. Only during a MODVAL (OP=R) are the VALINDX bits set to zero for deleted UIs. If there are several UN with the same UI (via FUI command), during a MODVAL (OP=R), then the VALINDX bit stays on, even if some of these UN are deleted. By not turning VALINDX bits off for deleted users, MODVAL can guarantee that new UN will not get a previously assigned (and then deleted) UI, until a MODVAL (OP=R) is run. This is protection of PFs.

When a UN is purged all pointers in LEVEL-0 and LEVEL-1 records are deleted, however, the LEVEL-2 record for this deleted user remain intact since the order of UN and UI have no bearing on which LEVEL-2 record will be used, this keeping of deleted users is no liability. During MODVAL (OP=R) these LEVEL-2 records are read and those with no LEVEL-1 pointers are eliminated, and if no other UN has that corresponding UI, all PFs for that UI are purged. No PFs are purged automatically until a restructure option is specified on MODVAL. At that time, MODVAL will purge all PFs assigned to any UI specified as unused in the VALIDUX file.

The VALINDX file consists of 4210B CM words or 44 sectors, 42 full data sectors, 1 short EOR sector, and one empty EOI sector. Each bit represents one of the 377777B UIs in the system. If the bit is on, UI is currently active. If it is off, UI is available for assignment.

The VALINDX file format is shown on the next page in Figure 11-2.

When MODVAL is called to create the VALIDUX and VALINDX files it reads the input cards and sets up all the UIs sequentially as they are read - unless a UI is specified. In call cases, MODVAL creates UI = 377777B for the SYSTEM, and UI = 377776B for LIBRARY. It sets up the VALINDX file with the bits set for those UIs it used and the file NEWVAL (VALIDUX) with all the level records required. Then control is passed to the UPDATE routine.



bit 59 word 0 of sector 1 is UI = 1

bit 0 word 0 of sector 1 is UI = 60D

bit 59 word 1 of sector 1 is UI = 61D

bit 0 word 1 of sector 1 is UI = 120D

Figure 11-2. VALINDX File Format

When MODVAL is called for UPDATE, it reads the input, the VALIDUX file, and VALIDNX file, and changes them accordingly. Then UPDATE writes the file either as changed or from information supplied from the create option.

If MODVAL is called via the K-display, any changes will cause an UPDATE. Since MODVAL works directly with the VALIDUX and VALINDX files, any changes are available as soon as K.END is typed for the entry changed.

The VALIDUX file is a two level tree file. Figure 11-3 shows the general method of progression through the tree into the required file information.

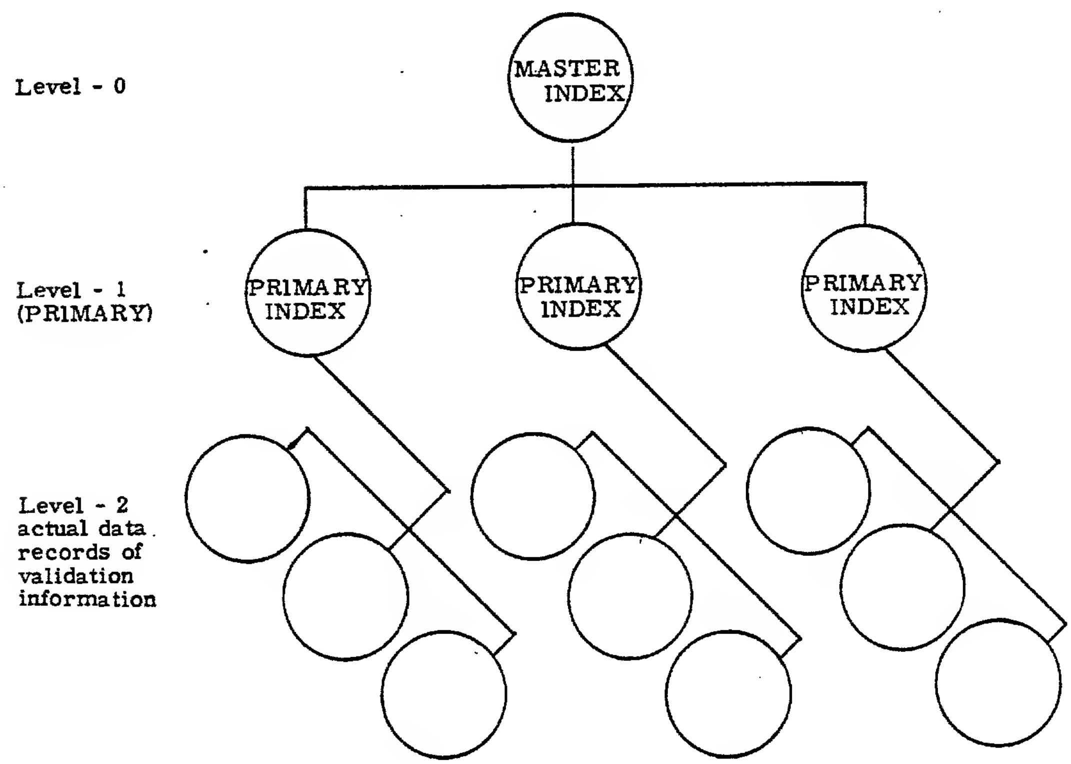


Figure 11-3. VALIDUX Tree File

The zero-level contains a fixed amount of data concerning the history of the file, and the first account number (and corresponding random index) of each Level - 1 block. The first PRU of this directory resides as sector one of the file and points to the next PRU of this directory.

The next level (primary) of the tree contains all validated account numbers with corresponding random addresses pointing to the Level - 2 blocks. Level - 2 blocks contain all the accounting information associated with this particular account number.

All Level - 1 records are less than one PRU in length or are linked through the control words if the data of a given level exceeds one PRU. Data for Level - 2 records must never exceed one PRU. Data in all levels is in alphabetical order (lowest item first). All Level Blocks have the same header word as last word of PRU. The header format is shown in Figure 11-4.

DL	WIB	WPE	NOE	FWAD

DL	=	Data Level
WIB	=	Words in Block
WPE	=	Words Per Entry
NOE	=	Number of Entries
FWAD	=	First Word Address of Data

Figure 11-4. VALIDUX, PROFILO Header Word

Figures 11-5 through 11-8 show the format for all levels of the file.

Control	0 [a]	n0 [b]	2 [c]	m0 [d]	3 [e]
	0		creation date	last MOD date	
	LW = RA of next Level - 0 block				
DATA 1	1st user number on this Level - 1 block *1				
	RA of above Level - 1 block				
DATA 2	1st user number on this Level - 1 block				
	RA of above Level - 1 block				
	• • •				
DATA n	1st user number on this Level - 1 block				
	RA of above Level - 1 block				

where:

a) Level number 0

b) There are n0 useful data CM words in this record exclusive of the 3 header words (actual numeric value)

c) There are 2 words per entry

d) There are m0 number of entries (actual numeric value)

e) Word number 3 of the record is the 1st entry on the record.

* Pointed by next word.

NOTE

The first word of the record is number 0, thus the fourth word of the record is really number 3. These may be short sectors.

Figure 11-5. Level-0 Block Format-VALIDUX

	a	b	c	d	e
Control	1	n1	2	m1	3
	0				
	LW = RA of next Level - 1 block				
DATA 1	user number				
	RA of the Level -2 which contains above user				
DATA 2	user number				
	RA of the Level - 2 block for above				
	⋮				
DATA n	user number				
	RA of the Level - 2 block for above				

where:

a) Level number 1

b) There are n1 useful data CM words (actual numeric value)

c) There are 2 words per entry

d) There are m1 number of entries (actual numeric value)

e) Word number 3 of the record is the 1st entry

NOTE

These are user numbers; they aren't associated with UI until Level - 2.

Figure 11-6. Level - 1 Block Format-VALIDUX

Whereas Level-0 block was a fast index to the Level-1 block, Level-1 is a direct pointer to the Level-2 block which contains the validation information for the specified user number.

	a	b	c	d	e
Control	2	n2	ARBS	m2	3
	0				
	0				
ARB1	1st word in ARB1 block for 1st user (account) number in this PRU				
	• • •				
	last word in ARB1 block for 1st user number in this PRU				
• • •	• • •				
ARBn	1st word in ARBn block for nth user number in this PRU				
	• • •				
	last word in ARBn block for nth user number in this PRU				

where:

a) Level number 2

b) There are n2 CM useful data words in this block

c) Each entry (ARB) is ARBS words long

d) There are m2(=4) entries in this block

e) The first entry begins in word number 3 of the block

Figure 11-7. Level -2 Block Format-VALIDUX

Currently Accounting Record Block Size (ARBS) = 15 so only 4 ARBS will fit in each block. When the search gets to this Level -2 record, all the information for validation is contained in one of the 4 ARB section of this block.

<table>
<tr><th>Offset Tag</th><th></th><th colspan="11"></th></tr>
<tr><td>ACCN</td><td>ARBn+0</td><td colspan="10">ACCOUNT NUMBER</td><td>USER INDEX</td></tr>
<tr><td>APSW</td><td>1</td><td colspan="10">PASSWORD</td><td>RESERVED</td></tr>
<tr><td>ACCD</td><td>2</td><td colspan="5">RESERVED</td><td colspan="5">CREATION DATE</td><td>LAST CHANGE DATE</td></tr>
<tr><td>AAB1</td><td>3</td><td colspan="11">ANSWER BACK CODE Number 1</td></tr>
<tr><td>AAB2</td><td>4</td><td colspan="11">ANSWER BACK CODE Number 2</td></tr>
<tr><td>AAB3</td><td>5</td><td colspan="11">ANSWER BACK CODE Number 3</td></tr>
<tr><td>AAB4</td><td>6</td><td colspan="11">ANSWER BACK CODE Number 4</td></tr>
<tr><td>APJN</td><td>7</td><td colspan="11">PROJECT Number of Account</td></tr>
<tr><td>AHMT</td><td>8</td><td colspan="4">MAX TAPES</td><td colspan="4">MAX PACKS</td><td>MAX TRACKS</td><td>MAX FILES</td><td>MAX DEFERRED BATCH JOBS</td></tr>
<tr><td>AHFC</td><td>9</td><td>R</td><td>RC</td><td>CS</td><td>FS</td><td>P</td><td>RO</td><td>X</td><td>TT</td><td>PRI</td><td>TIM</td><td>FL</td></tr>
<tr><td>AAWC</td><td>10</td><td colspan="11">ACCESS CONTROL WORD</td></tr>
<tr><td></td><td>11</td><td colspan="11">RESERVED</td></tr>
<tr><td></td><td>12</td><td colspan="11">RESERVED</td></tr>
<tr><td></td><td>13</td><td colspan="11">INSTALLATION AREA</td></tr>
<tr><td></td><td>14</td><td colspan="11">INSTALLATION AREA</td></tr>
</table>

where:

R	=	Reserved for System Use.
FC	=	Maximum Indirect Permanent File Count.
CS	=	Maximum Total Indirect File Space.
FS	=	Maximum Indirect File Size.
P	=	Terminal Parity.
RO	=	Number of Rubouts for Terminal User
X	=	Transmission Mode.
TT	=	Terminal Type.
PRI	=	Maximum CPU Priority.
TIM	=	Time Limit For CPU Program.
FL	=	Maximum Field Length.

Figure 11-8. Level-2 ARB Format-VALIDUX

The Common deck COMSACC has all the equivalences for using the VALIDUX file. The values are listed below for reference.

ACCN	EQU	0	Account number
AUIN	EQU	0	User index for account number
	VFD	42/ACCN	Account number
	VFD	18/AUIN	User index
APSW	EQU	1	Password for account
	VFD	42/APSW	Password for account
	VFD	18/OPEN	
ACCD	EQU	2	Coded creation date of account record
ACMD	EQU	2	Coded last modification date of account
	VFD	24/OPEN	
	VFD	18/ACCD	Coded creation date of account record
	VFD	18/ACMD	Coded last modification date
AAB1	EQU	3	Answer back code number one
	VFD	60/AAB1	Answer back code number one
AAB2	EQU	4	Answer back code number two
*	VFD	60/AAB2	Answer back number two
AAB3	EQU	5	Answer back code number three
	VFD	60/AAB3	Answer back number three
AAB4	EQU	6	Answer back code number four
	VFD	60/AAB4	Answer back number four
APJN	EQU	7	Project number of account
	VFD	60/APJN	Project number of account
AHMT	EQU	8	Highest no. of magnetic tapes allowed
AHRP	EQU	8	Highest number of removable packs allowed
AHMS	EQU	8	Highest number mass storage tracks allowed
AHNF	EQU	8	Highest number working files allowed
AHDB	EQU	8	Highest number deferred batch jobs allowed
	VFD	12/AHMT	Highest number of magnetic tapes
	VFD	12/AHRP	Highest number of removable packs
	VFD	12/AHMS	Highest number of mass storage tracks
	VFD	12/AHNF	Highest number of working files
	VFD	12/AHDB	Highest number of deferred batch jobs
AHFC	EQU	9	Highest number of indirect permanent files
AHCS	EQU	9	Highest number of indirect file sectors

AHFS	EQU	9	Highest number of sectors for one IA file
ATPA	EQU	9	Terminal parity
ATRO	EQU	9	Terminal rubouts
ATPX	EQU	9	Transmission mode
ATTT	EQU	9	Terminal type
AHPC	EQU	9	Highest priority for CPU allowed
AHTL	EQU	9	Highest time limit allowed
AHFL	EQU	9	Highest field length allowed in units of 100B octal words
	VFD	3/RESERVED	
	VFD	3/AHFC	Highest number of indirect permanent files
	VFD	3/AHCS	Highest number of indirect file sectors
	VFD	3/AHFS	Highest number of sectors for IA files
	VFD	1/ATPA	Terminal parity - may contain the following values:
APAE	EQU	0	Even parity
APAO	EQU	1	Odd parity
APAMX	EQU	2	(maximum number of values)
	VFD	5/ATRO	Terminal rubouts
			The following value has special meaning:
AROSY	EQU	37B	Use system default for terminal type
	VFD	1/ATPX	Transmission mode - may contain the following values:
APXH	EQU	0	Half duplex
APXF	EQU	1	Full duplex
APXMX	EQU	2	(maximum number of values)
	VFD	5/ATTT	Terminal type - may contain the following values:
ATTY	EQU	0	ASCII compatible terminal (TTY)
ATTC	EQU	1	Correspondence terminal
ATTCA	EQU	2	Correspondence, with APL character set
ATTMA	EQU	3	MEMOREX, with APL character set
ATTMX	EQU	4	(maximum number of values)
	VFD	12/AHPC	Highest priority for CPU
	VFD	12/AHTL	Highest time limit
	VFD	12/AHFL	Highest field length in 100B word units

Octal Value				
	AACW	EQU	10	Access control word
		ACCESS CONTROL WORD.		
		VFD	24/0	Installation area bits
		VFD	26/0	System reserved
		VFD	1/1	* CSTP* user special transaction privileges
		VFD	1/1	* CSRP* user may mount removable devices
		VFD	1/1	* CCNR* user may run without charge number
		VFD	1/1	* CAND* user assignment of non-alloc EQ
		VFD	1/1	* CASF* user access to system files
		VFD	1/1	* CSOJ* user can have system orgin, if debug
		VFD	1/1	* CSPF* user can create indirect access file
		VFD	1/1	* CLPF* user can create direct access file
		VFD	1/1	* CTPC* user can use telex privileged CMDS
		VFD	1/1	* CPWC* user can change his password
	DUMMA1	EQU	11	
		VFD	60/RESERVED FOR SYSTEM USE	
	DUMMA2	EQU	12	
		VFD	60/RESERVED FOR SYSTEM USE	
	DUMMA3	EQU	13	
		VFD	60/INSTALLATION AREA	
	DUMMA4	EQU	14	
		VFD	60/INSTALLATION AREA	
	AUIMX	EQU	377700B	Maximum UI for legal login or account card
	APFN	MICRO	1,, VALIDUX	Account validation file name
	AUFN	MICRO	1,, VALINDX	Available user index file name
7	AINC	EQU	7	Account number length in characters
20	.ANCR	EQU	AINC+9	Rounded account number length
1	AINW	EQU	.ANCR/10	Account number length in CM words
17	ARBS	EQU	15	Account record block size in CM words
4	ARBB	EQU	64/ARBS	Account records in Level-2 data block
2	ANWE	EQU	AINW+1	Account number words/entry

36	.ANPB	EQU	64/ANWE-2	Account number entries/blocks
34	ANLC	EQU	.ANPB/ARBB*ARBB	Less conrrol words
70	ABLI	EQU	ANLC*ANWE	Words per index block less control words
74	ABLB	EQU	ARBS*ARBB	Words per data block less control words

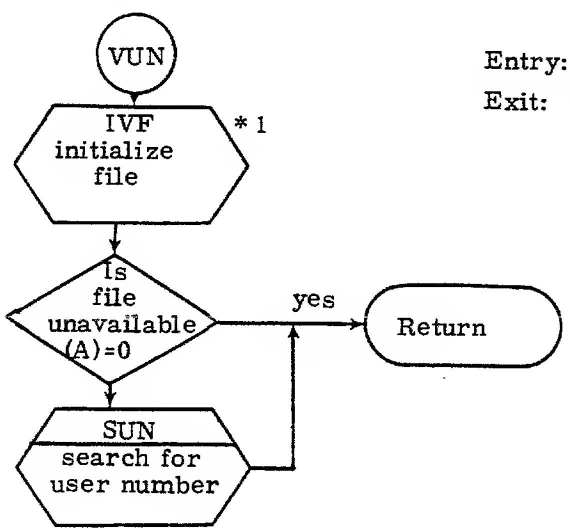
0AV is used by ACCFAM to locate the UI of a user number. Figure 11-9, the flow chart of 0AV shows how the tree files are used to locate an entry.

Basically, the Level-0 block is searched until the Account Number (AC) wanted is greater than the account number on the file (DATA). When this condition is found the RA of the last DATA points to the LEVEL - 1 block needed. (i.e. 1st DATA on RA n-1 > AC ≥ 1st DATA on RA n, then LEVEL-1 block is RA n-1).

The Level-1 block is searched until DATA=AC, then RA points to the Level-2 record.

The Level-2 block is searched until the account number field = AC, then this ARB is the information record (4 per Level-2 block). The password is then checked and UI is set accordingly. If, during the search, no match is found or some LINK RA is fradulent, UI is set to 0.

VUN - Main Program



Entry: Validation file attached

Exit: T1, T2 = UI if found
T1, T2 = 0 if not found
(A) = UI

Figure 11-9. 0AV Verify user Number

* 1 IVF - initialize validation file

ENTRY: (CN, . . . , CN+3) = Family name
= 0 if no family name

EXIT: (T4) = channel

(T5) = equipment

(T6) = first track

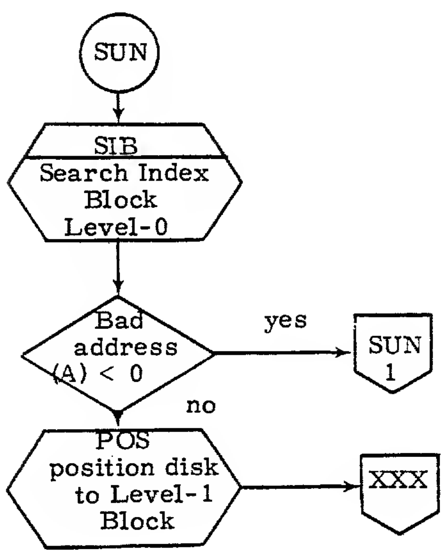
(T7) = first sector

(FTOV) = first track

channel will be reserved, file will be set busy

(A) = 0 if file unavailable

Figure 11-9. 0AV Verify User Number (Continued)



Entry: Validation file attached
(UN, . . . UN+3)=user number

Figure 11-10. SUN - Search for User Number

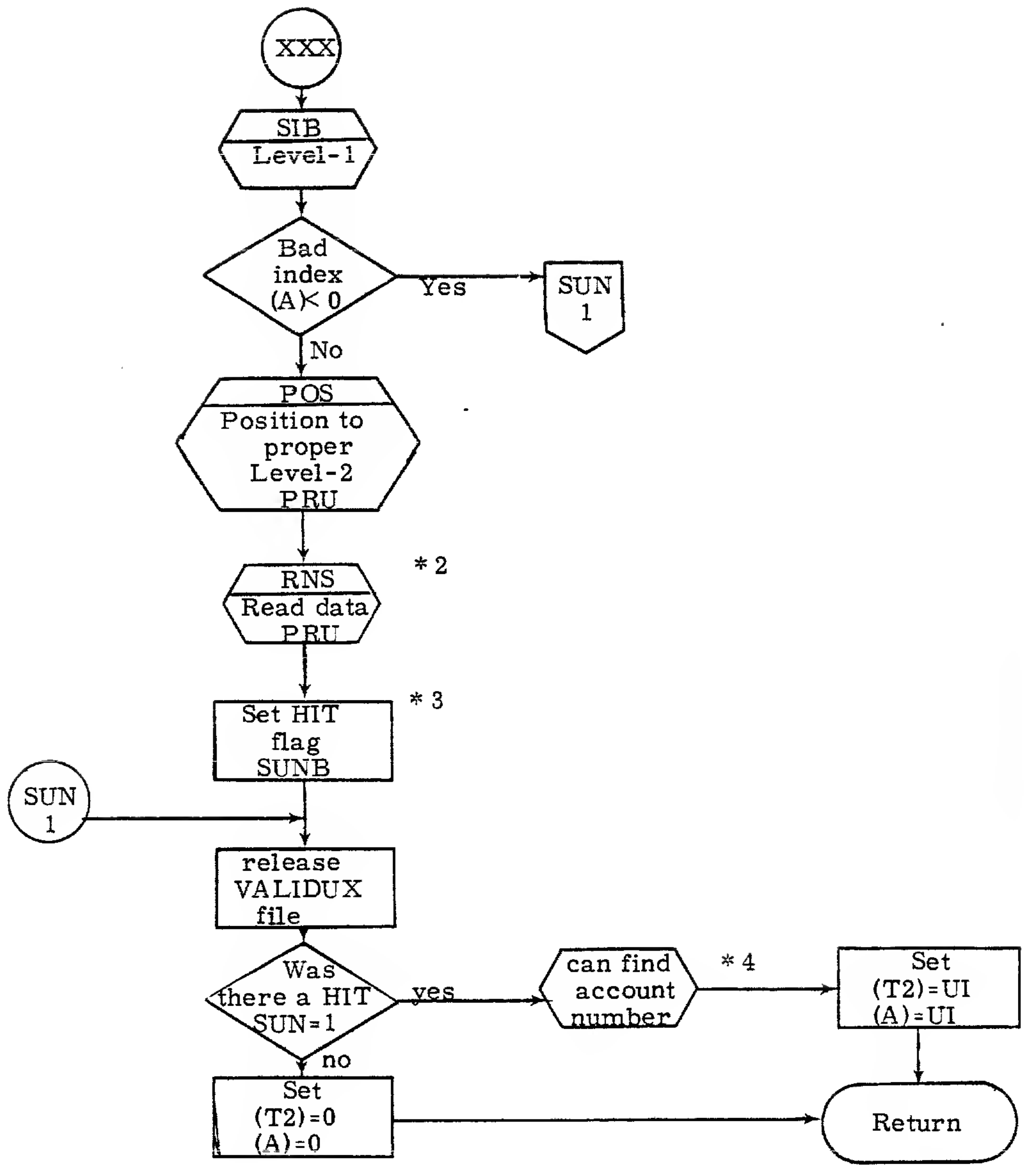


Figure 11-10. SUN - Search For User Number (Continued)

* 2 RNS - read next sector is a common deck COMPRNS.

ENTRY:	(A)	=	address of PP buffer
	(T4)	=	channel
	(T5)	=	equipment
	(T6)	=	track
	(T7)	=	sector
	channel reserved, disk positioned		
EXIT:	(A)	=	(T1) = word count of sector
	(T3)	=	address of PP buffer
	(T6 - T7)	=	advanced if no eoi

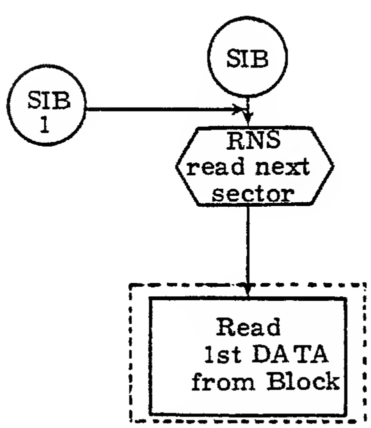
NOTE

It is known that this Level-2 record exists because if it did not, SIB would have returned (A)< 0 for a bad index.

* 3 instruction AOM SUNB will modify instruction SUNB LDN 0
NJN SUN6 IF HIT

to SUN LDN 1
NJN SUN6 IF HIT

* 4 Search Level-2 block for user and verify user for password number. If valid, set (A) = user index (UI). If not valid, set (A) = 0



Entry: (FTDV) = First track
disk positioned
(T6) = Track number
(T7) = Sector number
channel attached

Exit: (A)<0 if error (can't find entry)
(A)≥ 0 if no error

Figure 11-11. SIB - Search Index Block

Definitions:

AC = account number searching for

DATA = account number we found on file

Dashed lines indicate difference between Level-0 and Level-1 search

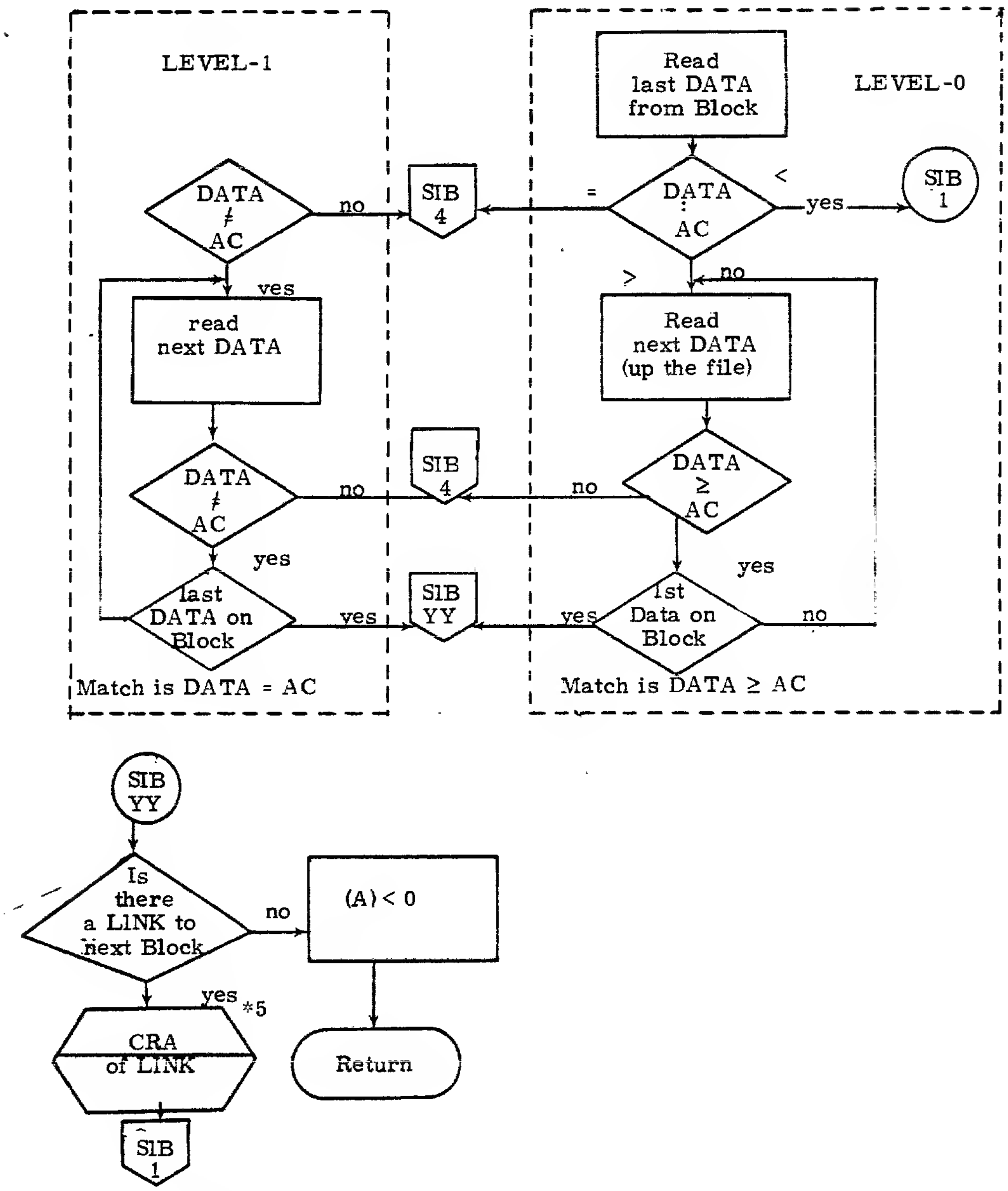
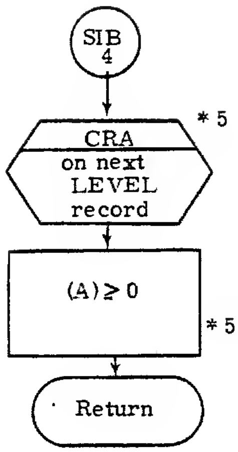


Figure 11-11. SIB - Search Block (Continued)



Found a Match:

- IF LEVEL-0 search then get RA of LEVEL-1 block

IF LEVEL-1 search then get RA of LEVEL-2 block

*5. CRA from common deck COMPCRA will convert random address (RA) to track and sector number.

ENTRY:	(T5)	= equipment number
	(T6)	= first track of file
	(RI, RI+1)	= RA
EXIT:	(A) < 0 if address out of bounds (i.e. not on the track chain)	
	(T6)	= track number
	(T7)	= sector number

NOTE

CRA must have the equipment number of the device to determine how many sectors per device.

Figure 11-11. SIB - Search Index Block (Continued)

11.4 PROFILE (PROFILO FILE)

PROFILE provides creation, modification and displays for the project profile file PROFILO.

PROFILE is a system utility which is used to create and maintain the special system file "PROFILO". PROFILO is a direct access permanent file residing under user index 377777B. PROFILO contains the information required to control a user's access to the system. The access is defined by charge, project, and user numbers, with additional limits on time-of-day and accumulated machine usage. The user is required to supply correct charge and account numbers if the "CCNR" bit in the users access word is clear. PROFILE also allows the definition of a master user for a charge number. This master user is validated to add or delete project numbers, account numbers, and user access information under the specified charge number. This modification to the charge number may be done under batch operation or from a TTY. A PROFILO file must exist for each permanent file family in the system, when this facility is desired. The PROFILE control card is:

PROFILE(P1-F1, ..., PN=FN, PM)

The control card is defined in the KRONOS 2.1 Reference Manual. Specific notes on the use of available options are:

1. When a new "PROFILO" file is generated under UI 377777B, ISF must be run to create a fast attach FNT entry for the file.

2. To attach the "PROFILO" file to a control point, ISF (R=PROFILO) must be run to release "PROFILO" from fast attach status.

3. When running PROFILE (OP=R) restructure file, the FNT entry for the fast attach file will be cleared. ISF must be run to reset an FNT entry.

11.4.1 PROFILO File Structure

The PROFILO file is a 3-level, tree-structured file and is accessed in the same manner as the VALIDUX file.

The PROFILO file can be updated either from the operators' console, batch or TTY.

The Level-0 contains a fixed amount of data concerning the history of the file, and the first charge number (and corresponding random index) of each Level-1 block. (Figure 11-12). The first PRU of this directory resides as Sector 1 of the file and points to the next PRU of this directory.

The next level (primary) of the tree contains all validated charge numbers with corresponding random addresses pointing to the Level-2 blocks.

A record in Level-2 of the tree contains all valid project numbers for the corresponding charge number. Along with each project number is a random address pointing to the Level-3 blocks.

Level-3 blocks contain all project profile information associated with this particular charge number and project number.

All records are less than one PRU in length and are linked through the control words if the data of a given level exceeds one PRU, with the exception that Level-3 records consisting of more than 1 PRU of data are on continuous PRUs. Data in all levels is in alphabetical order (lowest item first).

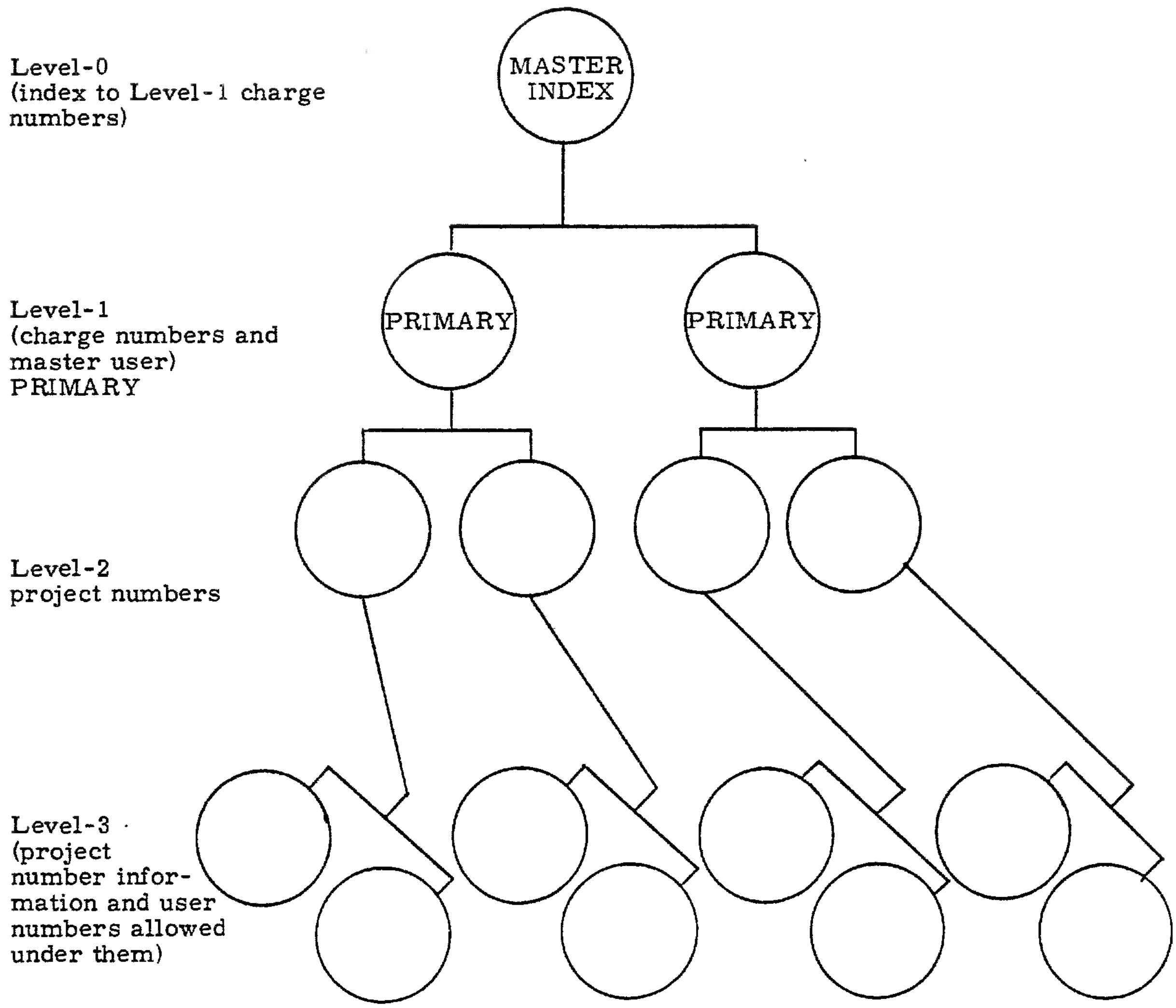


Figure 11-12. PROFILO Tree File Structure

Figures 11-13 through 11-16 give the format for the PROFILO file.

Control	a 0	b n0	c 2	d m0	e 3
	0		CREATION DATE		LAST MOD DATE
	LW = RA of next Level-0 record				
DATA 1	1st charge number in this Level-1 block				
	RA of above Level-1 block				
	• • •				
DATA n	1st charge number in this Level-1 block				
	RA of above Level-1 block				

Where:

a) Level number 0

b) There are n0 useful words in this block not counting the three header words

c) There are two words per entry

d) There are m0 entries in this block

e) Word number 3 is the 1st entry

Figure 11-13. Level-0 Block - PROFILO

	a	b	c	d	e	
Control	1	n1	3	m1	3	
	0					
	LW = RA of next Level-1 block					
DATA 1	charge number					
	master user number (ACCOUNT card user number)					* 1
	RA of Level-2 block for this charge number					
	• • •					
DATA n	charge number					
	master user number					
	RA of Level-2 block for this charge number					

Where:

a) Level number 1

b) There are n1 words of useful data in this block

c) There are 3 words per entry

d) There are m1 entries in this block

e) The 3rd word is the 1st entry

* 1 This associates the charge number with the master users UI

Figure 11-14. Level-1 Block - PROFILO

	a	b	c	d	e	
	2	n2	3	m2	3	
Control	0					
	LW = RA of next Level-2 for this charge number					
DATA 1	Project number part 1					
	Project number part 2					* 1
	RA of Level-3 block for this project number					
	• • •					
DATA n	Project number part 1					
	Project number part 2					
	RA of Level-3 block for this project number					

Each charge number has one or more Level-2 blocks associated with it. However, project numbers for only one charge number may reside in any one block. The charge number is not repeated since Level-1 points us to this block, so we know what charge number is associated with it.

* 1 If the project number is less than 10 characters, then this word is zero.

Figure 11-15. Level-2 Block - PROFILO

	a	b	c	d	e	
	3	n3	n3	1	PRUN	

RRPN	Project number part 1			
	Project number part 2			
PRCD	0	CREATION DATE	LAST MOD DATE	
PRTI	0	TIME IN	TIME OFF	* 1
PRCP	Maximum accumulated CPU time			* 2
PRAP	Accumulated CPU time			
PRCT	Maximum accumulated connect time			* 3
PRAT	Accumulated connect time			
	Open			
	Open			
	Open			
	Open			
	Open			
PRUN	User number 1 validated for this project number		0	* 4
	User number n-1		0	
	User number n		0	

a) Level number 3
b) There are n3 usable data words in this block
c) There are n3 data words in this entry (i.e. one entry per block).
d) There is one entry in this block.
e) The first user number is at word PRUN.

* 1 User may only use the system during the hours specified. As an example, if we set Time In (TI) to 08.00.00 and Time Off (TO) to 15.00.00, then this user can only run between 8 am and 3 pm.

Note

Level-3 records are not actually linked, but if they overflow they are written on successive PRUs. A PRUN user number of zero will end this Level-3 chain.

* 2 When PRAP is greater than PRCP user is no longer allowed to run.

* 3 When PRAT is greater than PRCT user is no longer allowed to run.

Note

PRAP and PRAT are not updated by the system as of release level 1.
If the master user wishes to keep the users on this project number from running on the system, he must set PRAP and/or PRAT in the PROFILO file. See the Installation Handbook for details on this procedure.

* 4 PRUN is the start of a list of all user numbers (account # from VALIDUX file) which are validated for this project number. Permanent files are assigned by the UI for this user number.

Figure 11-16. Level-3 Block - PROFILO

11.4.2 Equivalence Values For PROFILO FILE

Equivalence values for use with the PROFILO file are available in the common deck COMSPRO. A list of this equivalence is included here for reference.

Octal Value	Tag			Description
1	PRMU	EQU	1	Master user number
1	PRPN	EQU	1	Project number
3	PRCD	EQU	3	Creation and modification dates
4	PRTI	EQU	4	Time in
4	PRTO	EQU	4	Time off
5	PRCP	EQU	5	Maximum accumulated CP time
6	PRAP	EQU	6	Accumulated CP time
7	PRCT	EQU	7	Maximum connect time
10	PRAT	EQU	8	Accumulated connect time
16	PRUN	EQU	14	First user number
12	CINC	EQU	10	Charge number length in characters
23	.C	EQU	CINC+9	Rounded charge number length
24	PINC	EQU	20	Project number length in characters
35	.P	EQU	PINC+9	Rounded project number length
1	CINW	EQU	.C/10	Charge number length in CM words
2	PINW	EQU	.P/10	Project number length in CM words
13	CBNW	EQU	11	Profile control block length in CM words
2	CNWE	EQU	CINW+1	Charge number words/entry
3	CMWE	EQU	CNWE+1	Charge number + master U.N. words/entry
3	PNWE	EQU	PINW+1	Project number words/entry
40	CNPP	EQU	64/CNWE	Charge number entries per PRU
25	CMPP	EQU	64/CMWE	Charge number + master U.N. entries/PRU
25	PNPP	EQU	64/PNWE	Project number entries per PRU
36	CNLC	EQU	CNPP-2	Entries per PRU - control words
24	CMLC	EQU	CMPP-1	Entries per PRU - control words
—		EQU	64/PNWE	Protect number entries per PRU
24	PNLC	EQU	PNPP-1	Entries per PRU - control words
74	CBLP	EQU	CNLC* CNWE	Charge number block length per PRU
74	CMLP	EQU	CMLC* CMWE	Charge number + master U.N. block length/ PRU
74	PBLP	EQU	PNLC* PNWE	Project number block length per PRU
	PFFN	MICRO	1,,*PRO- FILO*	
	PPWD	MICRO	1,,*SE- CURUS*	PROFILO permanent file password
	PUSN	MICRO	1,,*SYS- TEMX*	PROFILO permanent file user number

Octal Value	Tag			Description
6	TOPT	EQU	6	Time-sharing update option
7	LOPT	EQU	7	List option
1001	BUFL	EQU	1001B	Input buffer length
101	PBUFL	EQU	101B	PROFILO buffer length
12	.CPB1	EQU	CNLC/3	
6	.CMB1	EQU	CMLC/3	
6	.PPB1	EQU	PNLC/3	
24	.CPB2	EQU	CNLC-.CPB1	
16	.CMB2	EQU	CMLC-.CMB1	
16	.PPB2	EQU	PNLC-.PPB1	
50	CPBP	EQU	.CPB2*CNWE	Charge number partial block length
52	CMPB	EQU	.CMB2*CMWE	Charge number + master U.N. partial block
52	PPBP	EQU	.PPB2*PNWE	Project number partial block length

The following is a dump of some MODVAL output and the VALIDUX, VALINDX files.

TABLE 11-4. MODVAL, VALIDUX, VALINDX

Foil No.	SE	Description
1 & 2	N/A	Listing of active users during a series of creating and deleting of user numbers.
3		DUMPTK of track number 264, which is the VALIDUX file.
3	1	LEVEL - 0 is 5 words in length. 3 header words and one DATA set since there is only 1 LEVEL-1 record. ALSON is the first user number alphabetically and the LEVEL - 1 record pointed to is RA = 3.
3	2	LIBRARY and SYSTEM are created first, then MLO and OPL were created. This is a LEVEL-2 record.
4	3	LEVEL-1 record is 45B words long, since there are 17D = 22B active users. Note: in the margins is indicated where words are zero and therefore non-existent in the dump, hence each sector has a 3 word header and each sector is a short sector, i.e., an EOR. The squared UN points to LEVEL-2 RA=4 on this slide. ALSON, USER, and USERALL are active; however, there is no pointer for DUMMY since it has been deleted.
4	4	This is a LEVEL-2 record. UN=ALSON UI=3, PW=ALSON See Figure 11-8 KRONOS 2.1 WORKSHOP manual for detail of the rest of this Account Record Block (ARB) UN=USER, UI=4, PW=N/A. UN=USERALL, UI=5, PW=USERALL. The last ARB has been deleted, hence the LEVEL-1 sector may be changed, but this ARB is not changed until a MODVAL (OP=R) is executed.
5		EOI sector
6		This is the VALINDX file.

TABLE 11-4. MODVAL, VALIDUX, VALINDX (Continued)

Foil No.	SE	Description
	1	The maximum UI used except for 377777B and 377776B is 50D = 62B, so only used 1st word. However, notice that all 44 sectors are allocated and are full sectors. UI=377777 and 377776 are not represented in VALINDX. This 1st word is shown on slide 7.
7		Breakdown of 1st word of VALINDX file representing UI=1 thru 60D.
7	44	EOI sector.

```
          MODVAL(OP=L,LO=N)                    74/07/18. 12.05.43.       PAGE     1

ACCOUNT     USER      CREATION   LAST MOD
NUMBER      INDEX       DATE       DATE

HLO             1     74/06/17.  74/07/01.
OPL             2     74/06/17.  74/06/17.
ALSON           3     73/06/19.  73/06/19.
USER            4     73/06/19.  73/06/19.
USERALL         5     73/06/19.  73/06/19.
DUMMY           6     74/07/18.  74/07/18.
DUMMY1          7     74/07/18.  74/07/18.
DUMM2          10     74/07/18.  74/07/18.
LIBRARY    377776     74/06/17.  74/06/17.
SYSTEMX    377777     74/06/17.  74/06/17.
```

1

```
          MODVAL(OP=L,LO=N)                    74/07/18. 12.06.45.       PAGE     1

ACCOUNT     USER      CREATION   LAST MOD
NUMBER      INDEX       DATE       DATE

HLO             1     74/06/17.  74/07/01.
OPL             2     74/06/17.  74/06/17.
ALSON           3     73/06/19.  73/06/19.
USER            4     73/06/19.  73/06/19.
USERALL         5     73/06/19.  73/06/19.
DUMM2          10     74/07/18.  74/07/18.
LIBRARY    377776     74/06/17.  74/06/17.
SYSTEMX    377777     74/06/17.  74/06/17.
```

2

```
          MODVAL(OP=L,LO=N)                    74/07/18. 12.07.41.       PAGE     1

ACCOUNT     USER      CREATION   LAST MOD
NUMBER      INDEX       DATE       DATE

HLO             1     74/06/17.  74/07/01.
OPL             2     74/06/17.  74/06/17.
ALSON           3     73/06/19.  73/06/19.
USER            4     73/06/19.  73/06/19.
USERALL         5     73/06/19.  73/06/19.
DUMM2          10     74/07/18.  74/07/18.
STUPID         11     74/07/18.  74/07/18.
HIGH           50     74/07/18.  74/07/18.
LIBRARY    377776     74/06/17.  74/06/17.
SYSTEMX    377777     74/06/17.  74/06/17.
```

3

```
           MODVAL(OP=L,LO=N)                    74/07/18. 12.08.57.      PAGE      1

ACCOUNT      USER      CREATION   LAST MOD
NUMBER       INDEX       DATE       DATE

MLO              1     74/06/17.  74/07/01.
OPL              2     74/06/17.  74/06/17.
ALSON            3     73/06/19.  73/06/19.
USER             4     73/06/19.  73/06/19.
USERALL          5     73/06/19.  73/06/19.
DUMM2           10     74/07/18.  74/07/18.
STUPID          11     74/07/18.  74/07/18.
ONE             12     74/07/18.  74/07/18.
TWO             13     74/07/18.  74/07/18.
THREE           14     74/07/18.  74/07/18.
HIGH            50     74/07/18.  74/07/18.
SIXTY           60     74/07/18.  74/07/18.
LIBRARY     377776     74/06/17.  74/06/17.
SYSTEMX     377777     74/06/17.  74/06/17.
```

(4)

```
           MODVAL(OP=L,LO=N)                    74/07/18. 12.09.51.      PAGE      1

ACCOUNT      USER      CREATION   LAST MOD
NUMBER       INDEX       DATE       DATE

MLO              1     74/06/17.  74/07/01.
OPL              2     74/06/17.  74/06/17.
ALSON            3     73/06/19.  73/06/19.
USER             4     73/06/19.  73/06/19.
USERALL          5     73/06/19.  73/06/19.
DUMM2           10     74/07/18.  74/07/18.
STUPID          11     74/07/18.  74/07/18.
ONE             12     74/07/18.  74/07/18.
TWO             13     74/07/18.  74/07/18.
HIGH            50     74/07/18.  74/07/18.
LIBRARY     377776     74/06/17.  74/06/17.
SYSTEMX     377777     74/06/17.  74/06/17.
```

(5)

```
           MODVAL(OP=L,LO=N)                    74/07/18. 12.11.04.      PAGE      1

ACCOUNT      USER      CREATION   LAST MOD
NUMBER       INDEX       DATE       DATE

MLO              1     74/06/17.  74/07/01.
OPL              2     74/06/17.  74/06/17.
ALSON            3     73/06/19.  73/06/19.
USER             4     73/06/19.  73/06/19.
USERALL          5     73/06/19.  73/06/19.
DUMM2           10     74/07/18.  74/07/18.
STUPID          11     74/07/18.  74/07/18.
ONE             12     74/07/18.  74/07/18.
TWO             13     74/07/18.  74/07/18.
FOUR            15     74/07/18.  74/07/18.
FIVE            16     74/07/18.  74/07/18.
SIX             17     74/07/18.  74/07/18.
SEVEN           20     74/07/18.  74/07/18.
HIGH            50     74/07/18.  74/07/18.
SIXTY1          61     74/07/18.  74/07/18.
LIBRARY     377776     74/06/17.  74/06/17.
SYSTEMX     377777     74/06/17.  74/06/17.
```

(6)

VALIDUX

DUMPTK(TK=264) LEVEL 0 RA = 1 DUMPTK - VER. 1 74/07/18. 12.11.43. PAGE 1

LEVEL2 RA = 2

WORD	TK=264 SE=0 B1=3777 B2=77		TK=264 SE=1 B1=2	B2=5		TK=264 SE=2 B1=3	B2=77 = 63D words
0	26011411042530001300	VALIDUX K	00000004000200010003	D B A C		00020076001700040003	B ¬ D D C
1	00004264114417013303	7‡19DA0C	00000010040621040722	HDFQDGR		00000000000000000000	
2	00000000040722140404	DGRLDD	00000000000000000000			00000000000000000000	
3	71261405600130033201	↓VLE≡AXCZA	01142317160000000000	ALSON		14110222012231377776	LIBRARY4‡¬
4	34023040051554007076	1BX⊃EM= ↑¬	00000000000000000003		C	00000000000000000000	
5	30431277310160033405	XBJ‡YA≡C1E	04251515350000000000	DUMM2		00000000040621040621	DFQDFQ
6	30071003160660400313	XGHCNF≡5CK	00000000000000000005		E	00000000000000000000	
7	14045400707637020603	LD= ↑¬4BFC	06112605000000000000	FIVE		00000000000000000000	
10	26011411042530377777	VALIDUX4‡‡	00000000000000000007		G	00000000000000000000	
11	01050000000242644000	AE B7‡5	06172522000000000000	FOUR		00000000000000000000	
12	00000000040621104645	DFQH-↓	00000000000000000007		G	00000000000000000000	
13	00000000030701131300	CGAKK	10110710000000000000	HIGH		00000000014400240001	A9 T A
14	00000000040621104645	DFQH-↓	00000000000000000005		E	00003700001277771000	4 J‡‡H
15	00000000000000000000		14110222012231000000	LIBRARY		77777777777777777777	‡‡‡‡‡‡‡‡‡‡
16	00000000000000003303	0C	00000000000000000002		B	00000000000000000000	
17	00000000000000000000		15141700000000000000	MLD		00000000000000000000	
20	70760510301404041070	↑¬EHXLDDH↑	00000000000000000002		B	00000000000000000000	
21	12010503000000013016	JAEC AXN	17160500000000000000	ONE		00000000000000000000	
22	34033007100316046010	1CXGHCND≡H	00000000000000000006		F	23312324051530377777	SYSTEMX4‡‡
23	37135400714730261074	4K= ↓‡XVH$	17201400000000000000	OPL		00000000000000000000	
24	22000017160134043111	R UNA1DYI	00000000000000000002		B	00000000040621040621	DFQDFQ
25	34063003100607333011	1FXCHFG0XI	23052605160000000000	SEVEN		00000000000000000000	
26	34063007100316203272	1FXGHCNPZ<	00000000000000000007		G	00000000000000000000	
27	54007177106323002100	= ↓‡HI5 Q	23113000000000000000	SIX		00000000000000000000	
30	54007176300610752100	= ↓¬XFH2Q	00000000000000000007		G	00000000000000000000	
31	66206010300612033414	VP≡HXFJC1L	23113024313400000000	SIXTY1		00000000000000000000	
32	50140010340637040563	/L HIF4DE‡	00000000000000000010		H	00000000014400240001	A9 7 A
33	30053201340530070100	XEZA1EXGA	23242520110400000000	STUPID		00003700001277771000	4 J‡‡H
34	70030100721754007231	↑CA <D= <Y	00000000000000000005		E	77777777777777777777	‡‡‡‡‡‡‡‡‡‡
35	14056001300332013402	LE≡AXCZA1B	23312324051530000000	SYSTEMX		00000000000000000000	
36	20007230310160033405	P <XYA≡C1E	00000000000000000002		B	00000000000000000000	
37	30071003160661707321	XGHCNF[↑>Q	24271700000000000000	TWO		00000000000000000000	
40	50007324137733105400	/ >TK‡0H=	00000000000000000006		F	00000000000000000000	
41	73240306370207443101	>TCF4BG9YA	25230522000000000000	USER		15141700000000000001	MLD A
42	60033405300310711251	≡C1EXCH↓J(	00000000000000000004		D	15141700000000000000	MLD
43	11500566300610060763	I/E∨XFHFG‡	25230522011414000000	USERALL		00000000040621040701	DFQDGA
44	30071003160560103010	XGHCNE≡HXH	00000000000000000004		D	00000000000000000000	
45	12340554300710031606	JIE*XGHCNF	00000000000000000000			00000000000000000000	
46	60101400340450047321	≡HL 1D/D>Q	00000000000000000000			00000000000000000000	
47	53040010054136041104	$D HE6301D	00000000000000000000			00000000000000000000	
50	05703005320134053007	E↓XEZA1EXG	00000000000000000000			00000000000000000000	
51	01007217000000000000	A <U	00000000000000000000			00040004175000620012	D DD/ 1 J
52	00000000010073262000	A >VP	00000000000000000000			07573700007777773244	G.4 ‡‡‡Z9
53	11130200131450001100	IKB KL/ 1	00000000000000000000			77777777777777777777	‡‡‡‡‡‡‡‡‡‡
54	05100200120316020320	EHB JCNBCP	00000000000000000000			00000000000000000000	
55	14020100131002006305	LBA KHB ‡E	00000000000000000000			00000000000000000000	
56	04720200635004061400	D<B ‡/DFL	00000000000000000000			00000000000000000000	
57	34571444010013103057	1.L9A KHX.	00000000000000000000			00000000000000000000	
60	60201701601056001312	≡POA≡H, KJ	00000000000000000000			17201400000000000002	OPL B
61	30131204055130201277	XKJDE(XPJ‡	00000000000000000000			23212501240000000000	SQUAT
62	34055100055160103010	1E(E(≡HXH	00000000000000000000			00000000040621040621	DFQDFQ
63	10060704140601001310	HFGDLFA KH	00000000000000000000			00000000000000000000	
64	02000547301410031620	B E*XLHCNP	00000000000000000000			00000000000000000000	
65	32725400460610632300	Z<= -FHI5	00000000000000000000			00000000000000000000	
66	21005400460530220504	Q = -EXRED	00000000000000000000			00000000000000000000	
67	14040100131030213406	LDA KHXQIF	00000000000000000000			00000000000000000000	

Note: This is an EOR

15D WORDS

15D WORDS

15D words

8D words next page

VALTDUX

```
        DUMPTK(TK=264)                                         DUMPTK - VER. 1     74/07/18. 12.11.43.   PAGE   2

 70     02004577300234273003  B +‡XB1WXC   00000000000000000000                  00040000014401440001  O  A9A9 A
 71     17013430060336303727  OA1XFC3X4*   00000000000000000000                  07273700007477771000  OW4   S‡‡H
 72     30213422140134233024  XQIRLA1SXT   00000000000000000000                  00000000000000000215            BH
 73     13771105342401007326  K‡1E1TA >V   00000000000000000000                  00000000000000000000
 74     00000000000006366322        F31H   00000000000000000000                  00000000000000000000
 75     03150010640443001063  CH H#D8 H1   00000000000000000000                  00000000000000000000
 76     67545055200106466300  A=/ PAF-1    00000000000000000000                  00000000000000000000
 77     43400043000106410677  85 ø AF6F‡   00000000000000000000                  00000000000000000000
          LEVEL 1  BLK   RA 3              LEVEL2   RA = 4                         LEVEL2   RA=5
WORD    TK=264  SF=3    B1=4    B2=45      TK=264  SE=4    B1=5    B2=77         TK=264  SE=5    B1=6    B2=77
  0     00010044000200210003   A ¥ B Q C   00020076001700040003  B - O D C       00020076001700040003  B - D D C
  3     01142317160000000000  ALSON        01142317160000000003  ALSON     C     00000000000000000000
  4     00000000000000000004           O   01142317160000000000  ALSON           04251515313400000007  DUMHY1   G
  5     04251515350000000000  DUMM2        00000000030623030623     CFSCFS       00000000040722040722     DORDOR
  6     00000000000000000005           E   00000000000000000000                  00000000000000000000
  7     06112605000000000000  FIVE         00000000000000000000                  00000000000000000000
 10     00000000000000000007           G   00000000000000000000                  00000000000000000000
 11     06172522000000000000  FOUR         00000000000000000000                  00000000000000000000
 12     00000000000000000007           G   00000000000000000000                  00000000000000000000
 13     10110710000000000000  HIGH         00040004014400240001  O DA9 T A       00000000014400240001       A9 T A
 14     00000000000000000005           E   07173700001277773720  004   J‡‡4P     00003700001277771000     4  J‡‡H
 15     14110227201223100000  LIBRARY      77777777777777777777  ‡‡‡‡‡‡‡‡‡‡      00000000000000000215            BM
 16     00000000000000000002           B   00000000000000000000                  00000000000000000000
 17     15141700000000000000  MLO          00000000000000000000                  00000000000000000000
 20     00000000000000000002           B   00000000000000000000                  00000000000000000000
 21     17160500000000000000  ONE          00000000000000000000                  00000000000000000000
 22     00000000000000000006           F   25230522000000000004  USER      O     04251515350000000010  DUMH2    H
 23     17201400000000000000  OPL          00000000000000000000                  00000000000000000000
 24     00000000000000000002           B   00000000030623030623     CFSCFS       00000000040722040722     DORDOR
 25     23052605160000000000  SEVEN        00000000000000000000                  00000000000000000000
 26     00000000000000000007           G   00000000000000000000                  00000000000000000000
 27     23113000000000000000  SIX          00000000000000000000                  00000000000000000000
 30     00000000000000000007           G   00000000000000000000                  00000000000000000000
 31     23113024313400000000  SIXTY1       00000000000000000000                  00000000000000000000
 32     00000000000000000010           H   00040004014400240001  D OA9 7 A       00000000014400240001       A9 T A
 33     23242520110400000000  STUPID       00003700001277771000    4  J‡‡H       00003700001277771000     4  J‡‡H
 34     00000000000000000005           E   00000000000000000215           BH     00000000000000000215            BM
 35     23312324051530000000  SYSTEMX      00000000000000000000                  00000000000000000000
 36     00000000000000000002           B   00000000000000000000                  00000000000000000000
 37     24271700000000000000  TWO          00000000000000000000                  00000000000000000000
 40     00000000000000000006           F   00000000000000000000                  00000000000000000000
 41     25230522000000000000  USER         25230522011414000005  USERALL   E     23242520110400000011  STUPID   1
 42     00000000000000000004           D   25230522011414000000  USERALL         00000000000000000000
 43     25230522011414000000  USERALL      00000000030623030623     CFSCFS       00000000040722040722     DORDGR
 44     00000000000000000004           D   00000000000000000000                  00000000000000000000
 51     00000000000000000000               000400040144002400D4  O DA9 T O       00000000014400240001       A9 T A
 52     00000000000000000000               07173700001277771000  GD4   J‡‡H      00003700001277771000     4  J‡‡H
 53     00000000000000000000               77777777777777777757  ‡‡‡‡‡‡‡‡‡.      00000000000000000215            BM
 60     00000000000000000000               00000000000000000000                  10110710000000000050  HIGH     /
 61     00000000000000000000               04251515310000000006  DUMMY     F     00000000000000000000
 62     00000000000000000000               00000000040722040722     DGROOR       00000000040722040722     DORDOR
 70     00000000000000000000    purged     00000000014400240001       A9 7 A     00000000014400240001       A9 T A
 71     00000000000000000000               00003700001277771000    4  J‡‡H       00003700001277771000     4  J‡‡H
 72     00000000000000000000               00000000000000000215           BH     00000000000000000215            BH
 77     00000000000000000000               00000000000000000000                  76330020000174046000  -0 P ASD≡
WORD    TK=264  SE=6    B1=7    B2=77      TK=264  SE=7    B1=10   B2=77         TK=264  SE=10   B1=11   B2=22
  0     00020076001700040003  B - O O C    00020076001700040003  B - O O C       00020021001700010003  B Q O A C
```

8D + 7D
= 15D
words

97404700C

VALTDUX

```
      DUMPTK(TK=264)                                               DUMPTK - VER. 1      74/07/18. 12.11.43.   PAGE   1

 3    17160500000000000012  ONE      J   06172522000000000015  FOUR     M   23113024313400000061  SIXTY1   1
 5    00000000040722040722     DGRDGR    00000000040722040722     0GRDGR   00000000040722040722     DGRDGR
13    00000000014400240001     A9 T A    00000000014400240001     A9 T A   00000000014400240001     A9 T A
14    00003700001277771000   4  J11H     00003700001277771000   4  J$$H    00003700001277771000   4  J$$H
15    00000000000000000215          BM   00000000000000000215          BM  00000000000000000215          BM
22    24271700000000000013  TWO      K   06112605000000000016  FIVE     N   00000000000000000006            F
23    00000000000000000000               00000000000000000000              17201400000000000000  OPL
24    00000000040722040722     DORDGR    00000000040722040722     0GRDGR   00000000000000000002            B
25    00000000000000000000               00000000000000000000              23242520110400000000  STUPID
26    00000000000000000000               00000000000000000000              00000000000000000005            E
27    00000000000000000000               00000000000000000000              23312324051530000000  SYSTEMX
30    00000000000000000000               00000000000000000000              00000000000000000002            R
31    00000000000000000000               00000000000000000000              24271700000000000000  TWO
32    00000000014400240001     A9 T A    00000000014400240001     A9 T A   00000000000000000006            F
33    00003700001277771000   4  J11H     00003700001277771000   4  J$$H    25230522000000000000  USER
34    00000000000000000215          BM   00000000000000000215          BM  00000000000000000004            D
35    00000000000000000000               00000000000000000000              25230522011414000000  USERALL
36    00000000000000000000               00000000000000000000              00000000000000000004            D
41    00000000000000000000               23113000000000000017  SIX      O   00000000000000000000
42    24102205050000000014  THREE    L   00000000000000000000              00000000000000000000
43    00000000040722040722     DGRDGR    00000000040722040722     DGRDGR   00000000000000000000
51    00000000014400240001     A9 T A    00000000014400240001     A9 T A   00000000000000000000
52    00003700001277771000   4  J11H     00003700001277771000   4  J$$H    00000000000000000000
53    00000000000000000215          BH   00000000000000000215          BH  00000000000000000000
60    00000000000000000000               23052605160000000020  SEVEN    P   00000000000000000000
61    23113024310000000060  SIXTY    ≡   00000000000000000000              00000000000000000000
62    00000000040722040722     DGRDGR    00000000040722040722     0GR0GR   00000000000000000000
70    00000000014400240001     A9 T A    00000000014400240001     A9 T A   00000000000000000000
71    00003700001277771000   4  J$$H     00003700001277771000   4  J$$H    00000000000000000000
72    00000000000000000215          BM   00000000000000000215          BH  00000000000000000000

WORD  TK=264  SE=11   B1=0    B2=0  2oi  TK=264  SE=12   B1=13   B2=100   TK=264  SE=13   B1=14   B2=100
 0    00000001463200223242  A-Z AZ7      00000000000000000165          A*  00000000000000000166          AV
 1    01532601141104253000  A$VALIDUX    77074071404076162011  $G5+55¬NP1   53436213771777041072  $B1K$D$DH<
 2    00240002763600000311   T B¬3  CI   26572561426130075162  V.U17(XG(1   21215226032025445706  QD)VCPU9.F
 3    17410003000022000307  U6 C  R CG   44174611764260035545  90-1¬7≡C +   70103441316523577107  +H16Y*5.+G
 4    01623327000146220001  A10* A-R A   35210675037710267757  2DF≥C$HV1.   71047203416515246140  +0<C6*HT15
 5    46320001511100015123  -Z A(1 A(5   11572255534553622521  1,R $+$)UD   43622432757424673476  R1TZ≥$T^1¬
 6    00010100100000030200   AA H  CB    20524130115446177656  P)6X1=-0¬,   50435150753444503032  /R(/219/XZ
 7    30703071307214771701  X+X+X<L$DA   64444444731042332325  #999>H705U   26217172710445350310  VQ+<+D+2CH
10    05763075605030511237  E¬X≥≡/X(J4   55164217201750545550   N70PO/* /   70577671044022274240  +.¬+D5RW75
11    10073474305113771006  HG1$X(K$HF   14054236632557214572  LE73$U.Q+<   73571735444120411051  >.0296P6H(
12    33500462100602000135  0/DJHFB A2   17517113565375550374  O(+K,$≥ C$   17650001053506053520  O* AE2FE2P
13    02000424011500055400  B 0TAH E=    50726577341346113776  /<*$1K-I4¬   61522543744541772247  [)UR$+61R*
14    01333001101431026114  A0X*HLYB[L   51356535243247142701  (2*2TZ*LWA   04753655544554540354  D≥3 **=*C=
15    37540100053610145400  4=A E3HL=    25354662560251641013  U2-1,B(≠HK   60711734122766470511  ≡+01JWV*E1
16    03261063230023005400  CVH$5 5 *    00620014236246602743   ) L51-≡W8   71125460450036225515  +J=≡+ 3R M
17    03252001301402000335  CUPAXLB C2   13170243430244277151  KOBBBB9W+(   55517357471507014537   (>.*MGA+4
20    05522002140102000335  E)P*LAB C2   35425725702711431653  27.U+WIBN$   57560670005122715722  .,F+ (R+.R
21    05041464617077745400  EOL≠[+1S=    15107242711427137560  MH<7+LWK≥≡   13562052747135605677  K,P)≤+2≡,$
22    02721702341750000533  B<0B10/ E0   72136332112375736012  <K$Z1S≥>≡J   74646502027140650020  $≠*BB+5* P
23    54000275500001355400  = B2/ A≥=    32600566027322526427  Z≡EvB>R)≠W   00355247437574254537   2)*B≥$U+4
24    03015000032554000305  CA/ CU= CE   03332264546214036530  COR≠=1LC*X   24012526637673375046  TAUV1¬>4/-
25    50000326540003061422  / CV= CFLH   31471706070050437147  Y*0FG /B+*   71571062460270131414  +.H1-B+KLL
26    02000364307660030200  B C*X¬≡CB    42723314555135602112  7<0L (2≡QJ   64274500474412606017  #W+ *9J≡≡0
27    05475600057603132000  E=, E¬CKP    61000444536304071422  [ 09$$0GLR   52606103377002267605  )≡[C4+BV¬E
30    04773517300710060607  D12UXGHFFG   25466151732436074461  U-[(>T3G9[   11017045212757553646  $A++QW. 3-
31    10713406140034070200  H+1FL 1GB    64621670342044416115  ≠1N+1P961M   40472351214766546147  5*S(0≠v=[*
```

11-38

VALINDX

DUMPTK(TK=265) DUMPTK - VER. 1 74/07/18. 12.13.52. PAGE 1

WORD	TK=265 SF=0 B1=3777 B2=77		TK=265 SE=1 B1=2 B2=100		TK=265 SE=2 B1=3 B2=100
0	26011411160430001200	VALINDX J	77777400000000401600	!!S DA≡	00000000000000000000
1	00004265114417016040	7*19DA≡5	00000000000000000000		00000000000000000000
2	00000000040722141515	DGRLHM	00000000000000000000		00000000000000000000
3	71261405600130033201	↓VL≤≡AXCZA	00000000000000000000		00000000000000000000
4	34023040051554007076	1BX⊃EM= ↑¬	00000000000000000000		00000000000000000000
5	30431277310160033405	XBJ!YA≡C1E	00000000000000000000		00000000000000000000
6	30071003160660400313	XGHCNF≡5CK	00000000000000000000		00000000000000000000
7	14045400707637020603	LD= ↑¬4BFC	00000000000000000000		00000000000000000000
10	26011411160430377777	VALINDX4!!	00000000000000000000		00000000000000000000
11	01050000000142654000	AE A7*5	00000000000000000000		00000000000000000000
12	00000000040621104645	DFQH-+	00000000000000000000		00000000000000000000
13	00000000040722141266	DGRLJV	00000000000000000000		00000000000000000000
14	00000000040621104645	DFQH-+	00000000000000000000		00000000000000000000
20	70760510301404041070	↑¬EHXLDDH↑	00000000000000000000		00000000000000000000
21	12010503000000013044	JAEC AX9	00000000000000000000		00000000000000000000
22	34033007100316046010	1CXGHCND≡H	00000000000000000000		00000000000000000000
23	37135400714730261074	4K= ↓*XVHS	00000000000000000000		00000000000000000000
24	22000017160134043111	R UNA1DY1	00000000000000000000		00000000000000000000
25	34063003100607333011	1FXCHFG0X1	00000000000000000000		00000000000000000000
26	34063007100316203272	1FXGHCNPZ<	00000000000000000000		00000000000000000000
27	54007177106343002100	= ↓!HIS Q	00000000000000000000		00000000000000000000
30	54007176300610752100	= ↓¬XFH≥Q	00000000000000000000		00000000000000000000
31	66206010300612033414	VP≡HXFJCIL	00000000000000000000		00000000000000000000
32	50140010340637040563	/L H1F4DE!	00000000000000000000		00000000000000000000
33	30053201340530070100	XEZA1EXGA	00000000000000000000		00000000000000000000
34	70030100721754007231	↑CA <D= <Y	00000000000000000000		00000000000000000000
35	14056001300332013402	LE≡AXCZA1B	00000000000000000000		00000000000000000000
36	20007230310160033405	P <AYA≡C1E	00000000000000000000		00000000000000000000
37	30071003160461707321	XGHCNF[↑>Q	00000000000000000000		00000000000000000000
40	50007324137733105400	/ >1K!0H=	00000000000000000000		00000000000000000000
41	73240306370207443101	>TCF4BG9YA	00000000000000000000		00000000000000000000
42	60033405300310711251	≡C1≤XCH↓J(	00000000000000000000		00000000000000000000
43	11500566300610060763	I/EVXFHFG!	00000000000000000000		00000000000000000000
44	30071003160560103010	XGHCNE≡HXH	00000000000000000000		00000000000000000000
45	12340554300710031606	J1E=XGHCNF	00000000000000000000		00000000000000000000
46	60101400340450047321	≡HL 1D/D>Q	00000000000000000000		00000000000000000000
47	53040010054136041104	$D HE63D1D	00000000000000000000		00000000000000000000
50	05703005370134053007	E↑XEZ41EXG	00000000000000000000		00000000000000000000
51	01007217000000000000	A <U	00000000000000000000		00000000000000000000
52	00000000010073262000	A >VP	00000000000000000000		00000000000000000000
53	11130200131450001100	IKB KL/ I	00000000000000000000		00000000000000000000
54	05100200170316020320	EHB JCNBCP	00000000000000000000		00000000000000000000
55	14020100131002006305	LBA KHB !E	00000000000000000000		00000000000000000000
56	04720200635004061400	D<B !/DFL	00000000000000000000		00000000000000000000
57	34571444010013103057	1.LYA KHX.	00000000000000000000		00000000000000000000
60	60201701601056001312	≡PDA≡H, KJ	00000000000000000000		00000000000000000000
61	30131204055130201277	XKJDE(XPJ!	00000000000000000000		00000000000000000000
62	34055100055160103010	1E(E(≡HXH	00000000000000000000		00000000000000000000
63	10060704140601001310	HFGDLFA KH	00000000000000000000		00000000000000000000
64	02000547301410031620	B E*XLHCNP	00000000000000000000		00000000000000000000
65	32725400460610632300	Z<= -FHIS	00000000000000000000		00000000000000000000
66	21005400460530220504	Q = -EXRED	00000000000000000000		00000000000000000000
67	14040100131030213406	LDA KHXQ1F	00000000000000000000		00000000000000000000
70	02004577300234273003	B ↓!XH1WXC	00000000000000000000		00000000000000000000
71	17013430060336303727	OA1AFC3X4W	00000000000000000000		00000000000000000000
72	30213422140134233024	XQ1HLA1SXT	00000000000000000000		00000000000000000000

97404700C

VALINDX

```
DUMPTK(TK=265)                                              DUMPTK - VER. 1      74/07/18. 12.13.52.    PAGE   2

 73    13771105342401007326  K↑I61TA >V     00000000000000000000                   00000000000000000000
 74    00000000000006366322        F31R     00000000000000000000                   00000000000000000000
 75    03150010640443001063  CH H≠DB H↑     00000000000000000000                   00000000000000000000
 76    67545055200106466300  A=/ PAF=↑      00000000000000000000                   00000000000000000000
 77    43400043000106410677  85 8 AF6F↑     00000000000000000000                   00000000000000000000

WORD   TK=265  SE=44  B1=0   B2=0  eol  TK=265  SE=45  B1=46  B2=100        TK=265  SE=46  B1=47  B2=100
 0
 1
 2
 3
 4
 5
 6
 7
10
11
12
13
14
15
16
17
20
21
22
23
24
25
26
27
30
31
32
33
34
35
36
37
40
41
42
43
44
45
46
47
50
51
52
53
54
55
56
57
60
61
62
```

7 7 7 7 7 4 0 0 0 0 0 0 4 0 1 6 0 0 0

OCTAL

DECIMAL

VALINDX had the following UI's active

1	11	50	60
2	12		61
3	13		62
4	14		
5			
6			
7			
10			

circled UI's have been PURGED

LOADER 12

12.0 INTRODUCTION

There are actually two loaders. One is a PP routine, LDR, (also called LDV) which is an absolute CM routine loader. This LDR loads absolute binaries directly from the RCL or directly from the system RMS device, or from a local file. The second loader is the relocating loader, LINK. LINK takes a relocatable type binary deck and absolutizes it at whatever location in a CP's CM desired.

For completeness, the PP loader and Alternate System Residency (ASR) are mentioned here. The PP routine loader, PLL, is discussed in Section 4, and ASR is discussed in this section.

The basic flow of loading central memory programs is detailed in Section 8 of the KRONOS 2.1 Reference Manual. All the loader table formats are presented in Appendix D of the Reference Manual. Subsequent discussions in this section are a supplement to the KRONOS 2.1 Reference Manual.

The Loaders can be called in the following ways.

1) A control card call, LOAD
2) A control card, such as COMPASS, LGO, etc.
3) An RA+1 call to LDR to process overlay loading (overlays are always absolute routines).

12.1 LDR (LDV)

The LDR call is shown in Figure 12-1.

LDR Calls from CPU Routine

	18			
RA+1	LDR	R	O	addr

R - auto recall if desired

addr - address of request

Figure 12-1. LDR Call

The load request consisting of 2 or 4 words is shown in Figure 12-2.

<table>
<tr><td>addr+0</td><td colspan="8">NAME</td><td></td><td>0</td><td></td></tr>
<tr><td></td><td>6</td><td>6</td><td>2</td><td>3</td><td>1</td><td>1</td><td>4</td><td>1</td><td>18</td><td>18</td><td>No. of Bits</td></tr>
<tr><td>addr+1</td><td>L1</td><td>L2</td><td>N</td><td>0</td><td>U</td><td>V</td><td>0</td><td>E</td><td>LWA</td><td>FWA</td><td>Name</td></tr>
<tr><td>addr+2</td><td colspan="9">OVLNAME</td><td>0</td><td></td></tr>
<tr><td>addr+3</td><td colspan="9">EPTNAME</td><td>0</td><td></td></tr>
</table>

NAME	-	Source of name depending on U and N.
L1	-	First overlay level.
L2	-	Second overlay level.
N	-	Number of words in request - 2.
U	-	Load option (see below).
V	-	Overlay flag (Must be set to 1).
E	-	Call completion flag (see below).
FWA	-	First word address of overlay.
LWA	-	Last word address available for load
OVLNAME	-	Name of overlay to be loaded (if N≠0).
EPTNAME	-	Entry point name when loading multiple entry point overlay (if N=2).

Note.

If U=0, N is ignored and NAME is the name of the file containing the overlay. (L1 and L2 are required).

If U=1 and N=0, NAME is the name of the overlay from the system, (L1 and L2 are ignored).

If U=1 and N≠0, OVLNAME is the name of the overlay from system (L1 and L2 are ignored).

If FWA=0, Overlay is loaded at address specified by overlay.

If L1=L2=0, Control is to called overlay, otherwise, control is returned to caller with FWA = entry address.

If E=1, Control transfers to the specified entry point (EPTNAME) in the overlay.

Figure 12-2. Load Request

Upon completion of the load, information is returned in the call block shown in Figure 12-3.

				18
addr+0	NAME			0
addr+1	6 L1	6 L2	12 0	EPTADDR
addr+2	OVLNAME			0
addr+3	EPTNAME			0

where:

EPTADDR - Entry point address of overlay

If N=2
EPTADDR = Address of EPTNAME

Figure 12-3. Load Completion Call Block

Dayfile messages associated with Figure 12-3 include:

1. OVERLAY NOT FOUND IN LIBRARY - Requested overlay was not found in the system library.
2. ARG ERROR - LDR parameters were outside FL.
3. FILE NOT OVERLAY FORMAT - First record of file was not an overlay.
4. LDR ERROR - Issued before one of preceding errors.

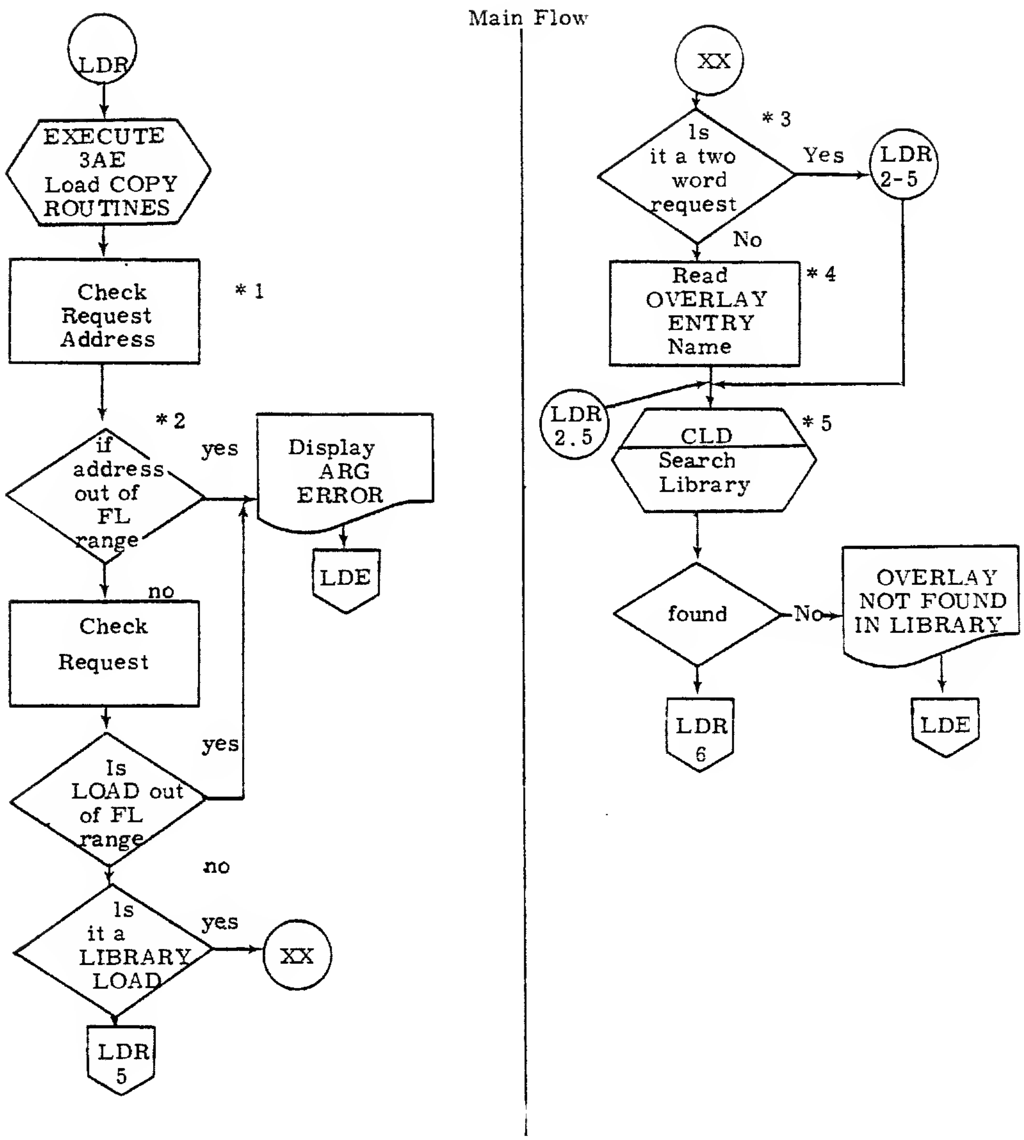
LDR will load its copy routines from 1AJ overlay 3AE. These are a group of subroutines used to load CM programs which consist of:

1. LCP - Load Central Program
2. CMS - Copy Mass Storage-resident program
3. SLP - Set load parameters
4. CSF - Check Special Format (ACPM)
5. CCM - Load (ECS) resident programs (ASR)
6. Several format checking routines (ACPM Table)

LDR will then check for proper argument program load not out of bound, etc.) and load the program or overlay. When completed, LDR drops and, if recall was used, CPUMTR will let the caller continue. A flowchart of the Loader loop is shown in Figure 12-4.

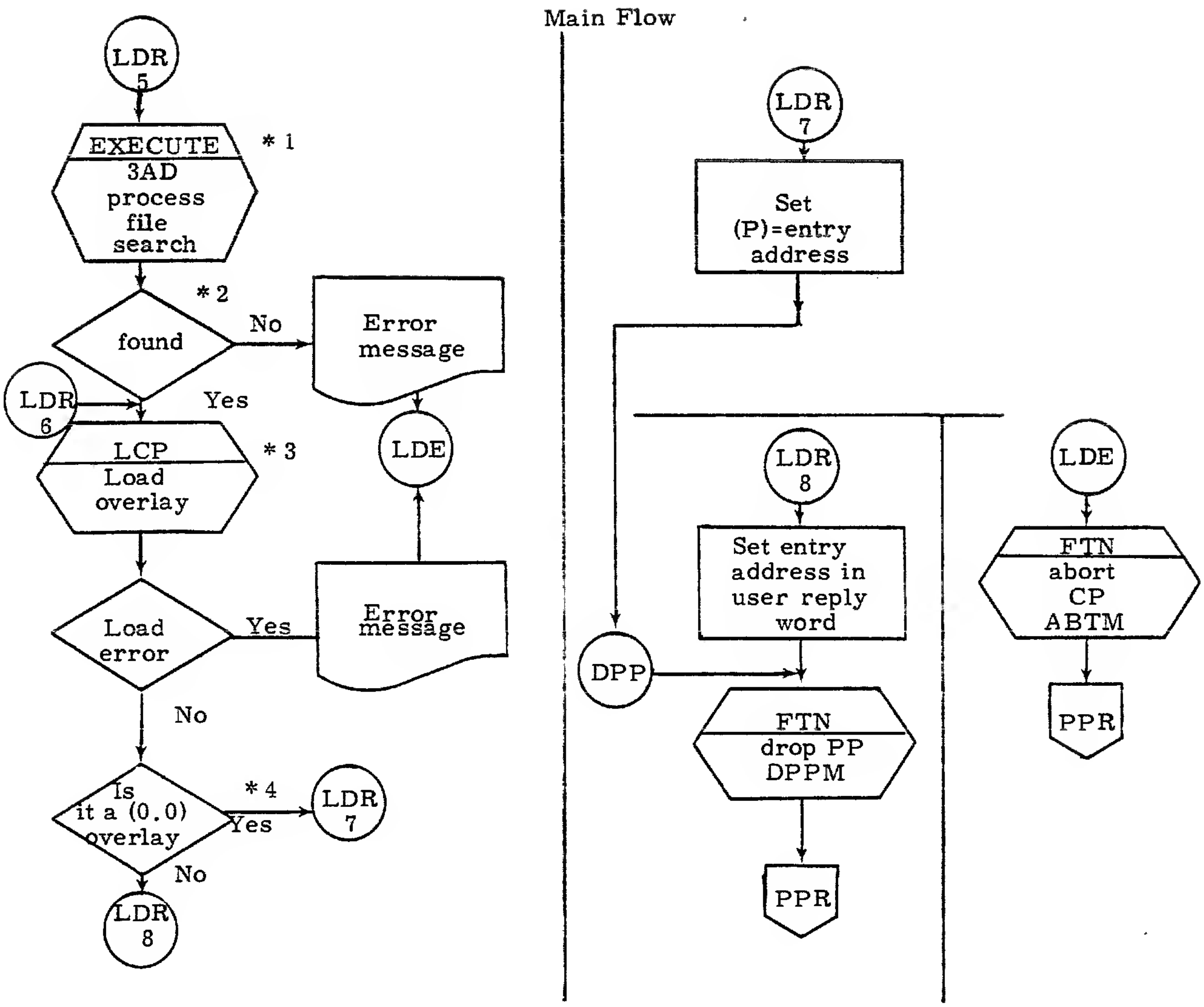
NOTE:

LDR automatically starts a (0,0) level overlay regardless of the instructions specified in the LDR control block. For this reason, if a (0,0) level overlay is to be loaded, but not started automatically, the user can use a READSKP or READR CIO request. A (0,0) overlay load is enterpreted by LDR as a request to overlay the caller of LDR.



* 1 is lower 18 bits of RA+1 call out of users FL.
* 2 is location to load program out of users FL.
* 3 this is not an OVERLAY call.
* 4 get word 3 & 4 of call.
* 5 Common deck COMPCLD.

Figure 12-4. LDR - ABS, COS, and OVL Loader



* 1	3AE and 3AD reside in core with LDR. See 1AJ overlay core layout in Section 6. 3AD uses common decks COMPSAF-search for assigned file and COMPSFB-set file busy. 3AD searches Library for the load name.
* 2	This check is actually performed in overlay 3AD
* 3	In overlay 3AE.
* 4	or a main routine.

Figure 12-4. LDR - ABS, COS, and OVL Loader (continued)

12.2 LINK (RELOCATABLE LOADER)

LDR will load LINK at RA+100. LINK will process the binary file and begin building an absolute deck relocated from RA+100 (or other load address if specified) at the end of the code. Link will build reference tables and any other tables needed at the end of the user's FL. The program will grow downward and tables will grow upward. If they meet, the FL is too small to load the routine (See Figure 12-5). In the last 20D words of FL there is a small move and preset routine. LINK will build the absolute program unitl it reaches EOR on the file. Then it will use the tables it built to satisfy all local transfers. It then reads the next routine on the file, loading it behind the previous one. When LINK finally encounters an EOF or EOI on the file, it attempts to link all the loaded routines together and satisfy all external references from any libraries specified. If undefined external references remain, it links them to an object time routine LDRUSX, which will abort the control point if called.

If the execution flag is set (LGO card EXECUTE card, etc.), LINK will transfer to the last 20 locations in this F. This small routine will move the absolute code produced by LINK to its load point (normally RA+100). If Preset is selected, core is set and control is transferred to the transfer address specified in the transfer table, XFER. In addition if the execution flag is not set, LINK will read the next control card. If it is not one of its control cards, it will go inactive, and 1AJ will reprocess the control card.

The control cards LINK processes are:

1. LDSET
2. LOAD
3. MAP
4. NOMAP
5. LIBRARY
6. REDUCE
7. SETCORE

In addition, for debug purposes the console operator under the DIS package can type RSS.LGO (if the binaries exist on LGO, if not, any lfn can be used). This command will force LINK to load the binaries in the usual manner and set P = transfer address from the XFER table. It will also execute the move and preset core loop, but the loaded routine will not be put into execution. The move loop will drop the CPU and the operator may break point.

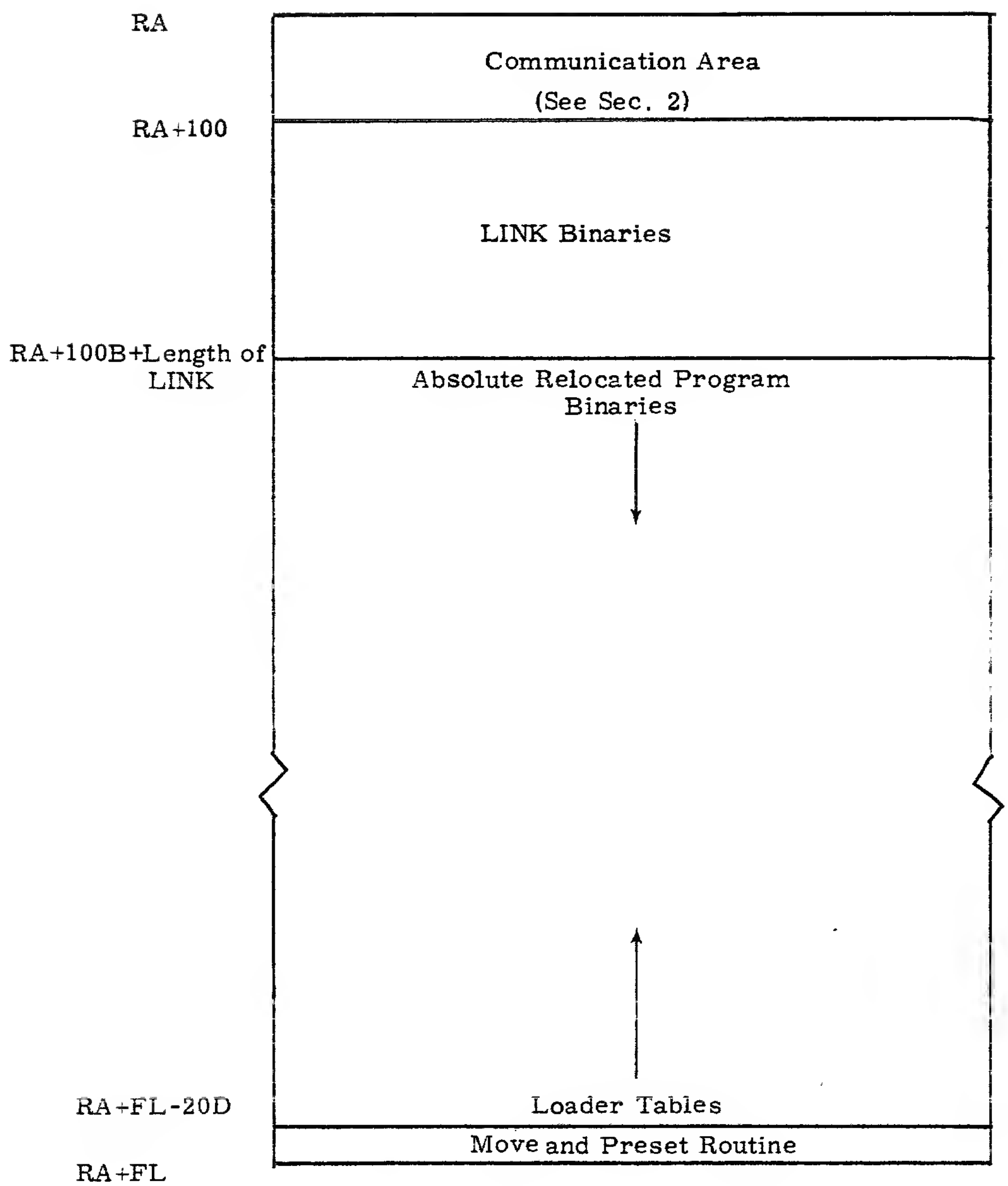


Figure 12-5. Core Layout of LINK

If the control card which forces a call to LINK, is an LGO relocatable type program call. LDR will find that the binary deck is relocatable (either a 70-LDSET or 34-PIDL type table follows the 77-IDENT type table). It will load the overlays as shown in Figure 12-6 (level-1) and set the Execution Flag (XF) On. LDR will set (P)="LDR=". LINK will then begin processing the binary file.

If the control card which forces a call to LINK is a LDSET, LOAD, CARD, etc., LDR will simply load the overlay, LINKCTL, and will set (P)=control card name. LINKCTL will process the card. If the card was an EXECUTE card, then LINK will have LDR= loaded, set the execution flag, XF, On, and go to CPL. Each card has its own entry point.

If LDR= detects an OVERLAY directive in the binary, it will have OVG= overlay loaded and will let OVG= process these binaries. The OVERLAY directive must be the 1st card of the overlay, otherwise the load is aborted. If the control card call was a LIBGEN card, LDR loads just the overlay, LIBGEN, as shown in Figure 12-6 (level -2) and will set (P)=LIBGEN.

If the control card call was a LINK card, the overlay, LINKLNK, is loaded and (P)=LINK is set as shown in Figure 12-6 (Level -3).

All of the tables generated and used by the loader are dynamic and controlled by the use of three macros and their associated routines.

1. ADDWRD and ADW - add 1 word to managed table.
2. SCAN and STE - scan table for entry.
3. ALLOC and ATS - allocate table space.

NOTE

> Similar table management macros can be found in the Common Deck COMCMTM and the associated routines are in the Common Deck COMCMTP. Also reference Appendix A of the KRONOS 2.1 Reference Manual.

In addition, one other macro, TABLE is used to generate a managed table.

By using these macros, all the tables will expand and contract without consciencious effort of the code of LINK which uses them.

Figure 12-7 flowcharts the general flow in level-1.

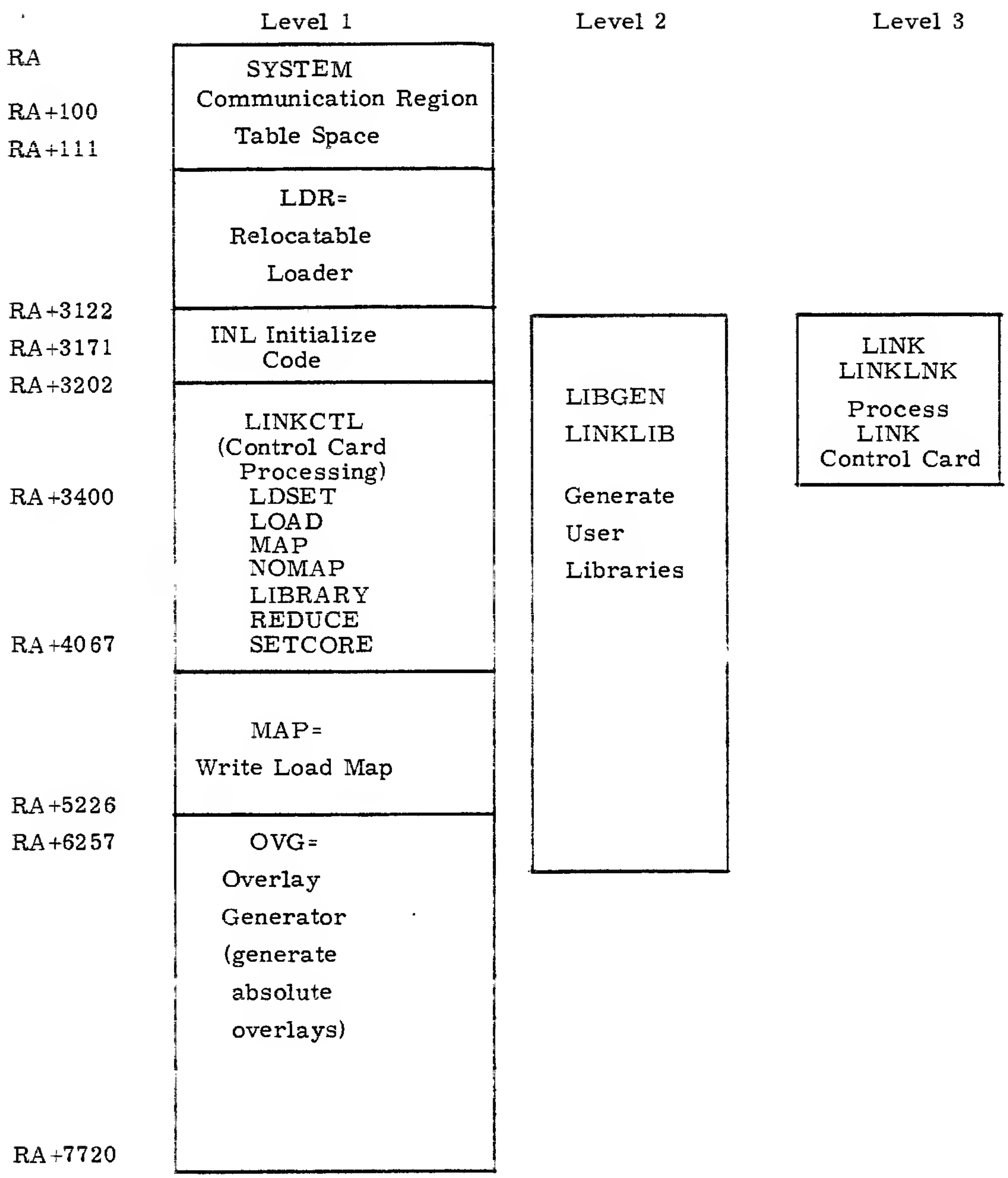


Figure 12-6. LINK Overlay Core Structure

LDR loads LINK and the transfer table specifies LDR= as entry point. First executable code for LINK.

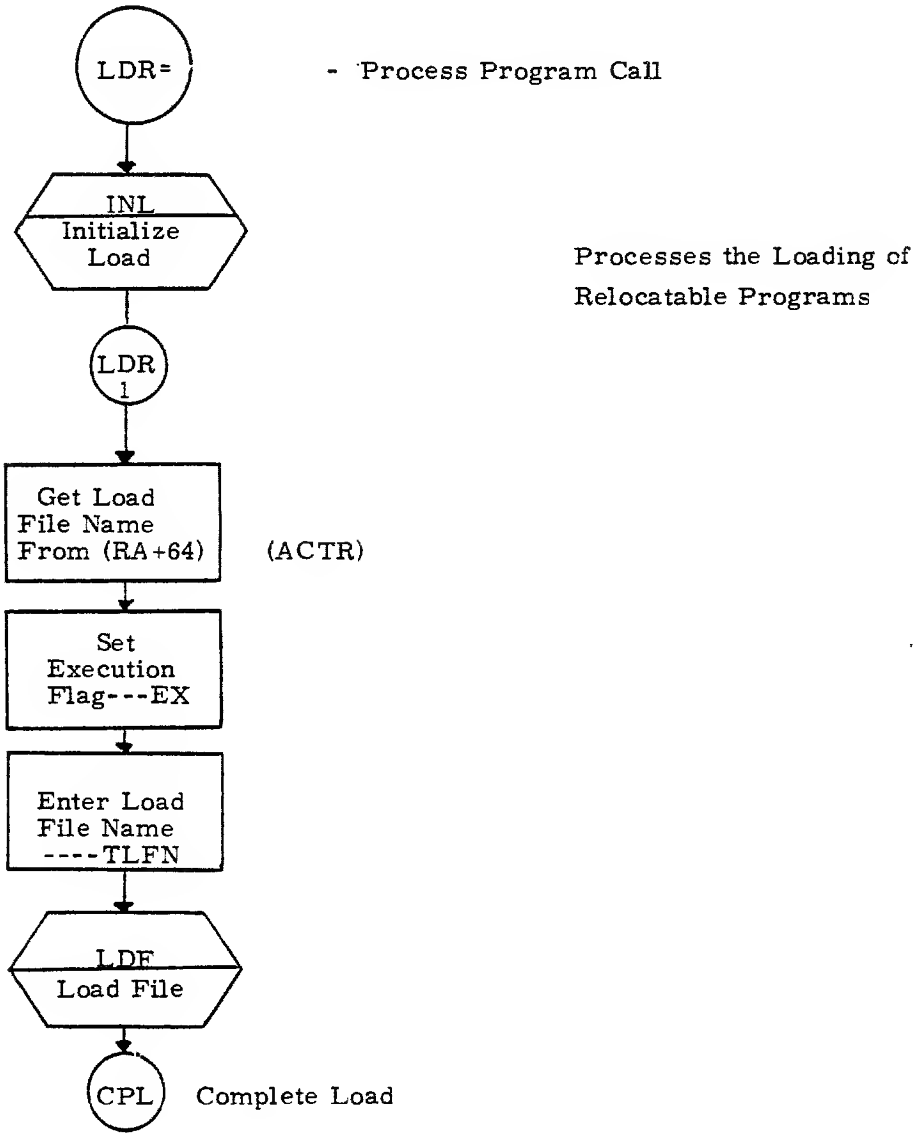


Figure 12-7. LINK Flow In Level-1

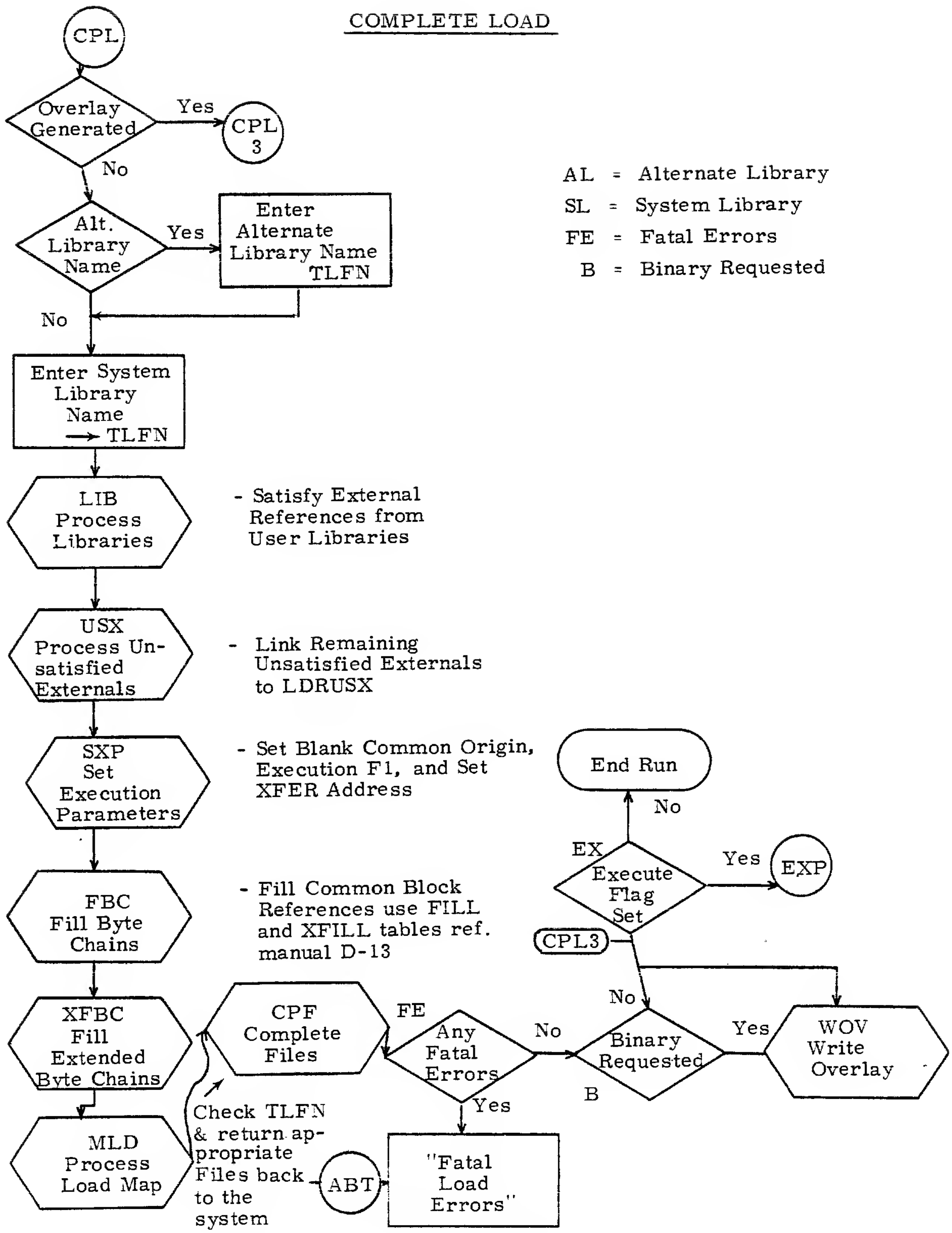


Figure 12-7. LINK Flow in Level -1 (Continued)

EXECUTE PROGRAM

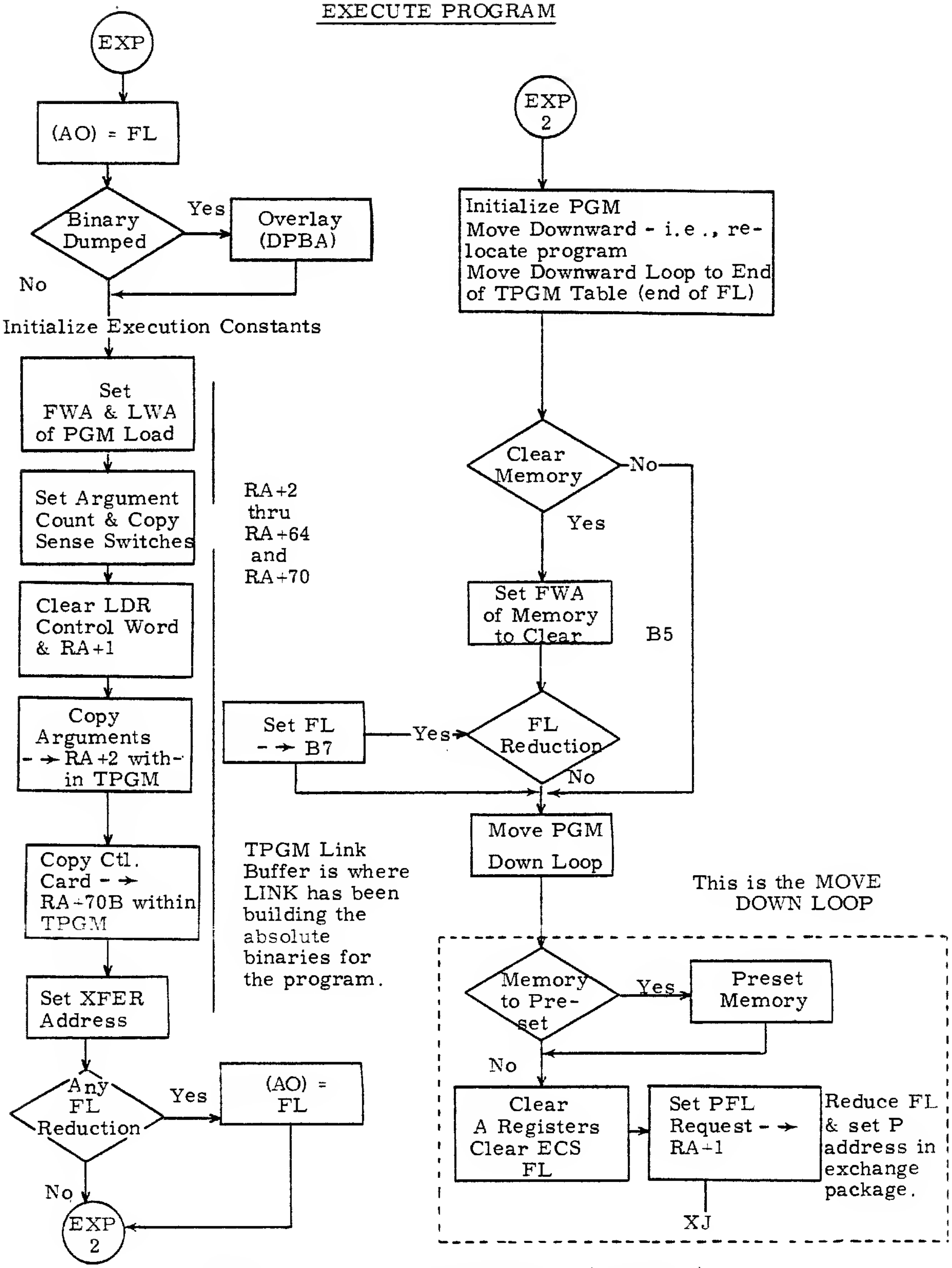


Figure 12-7. LINK Flow in Level-1 (Continued)

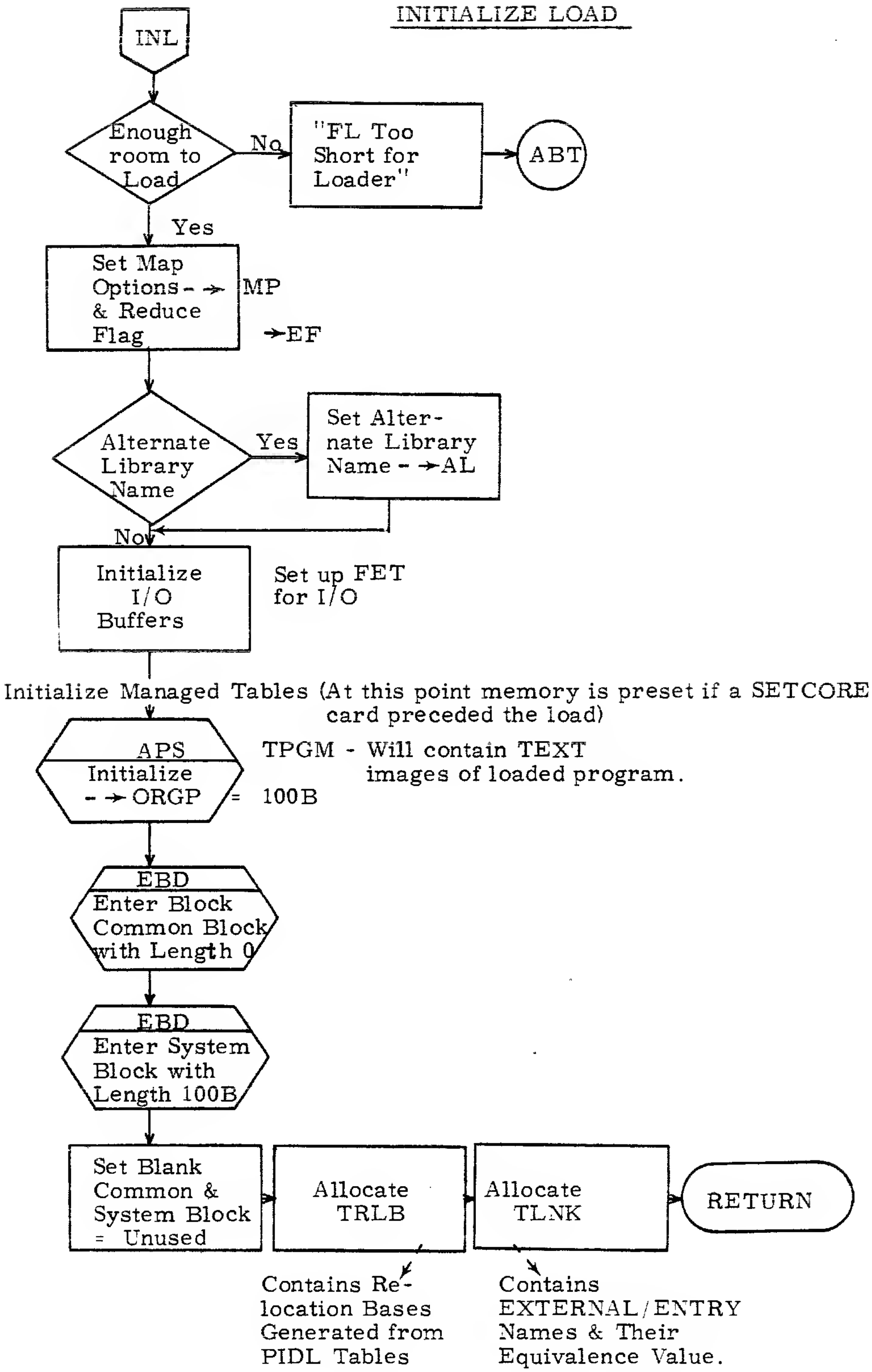


Figure 12-7. LINK Flow in Level-1 (Continued)

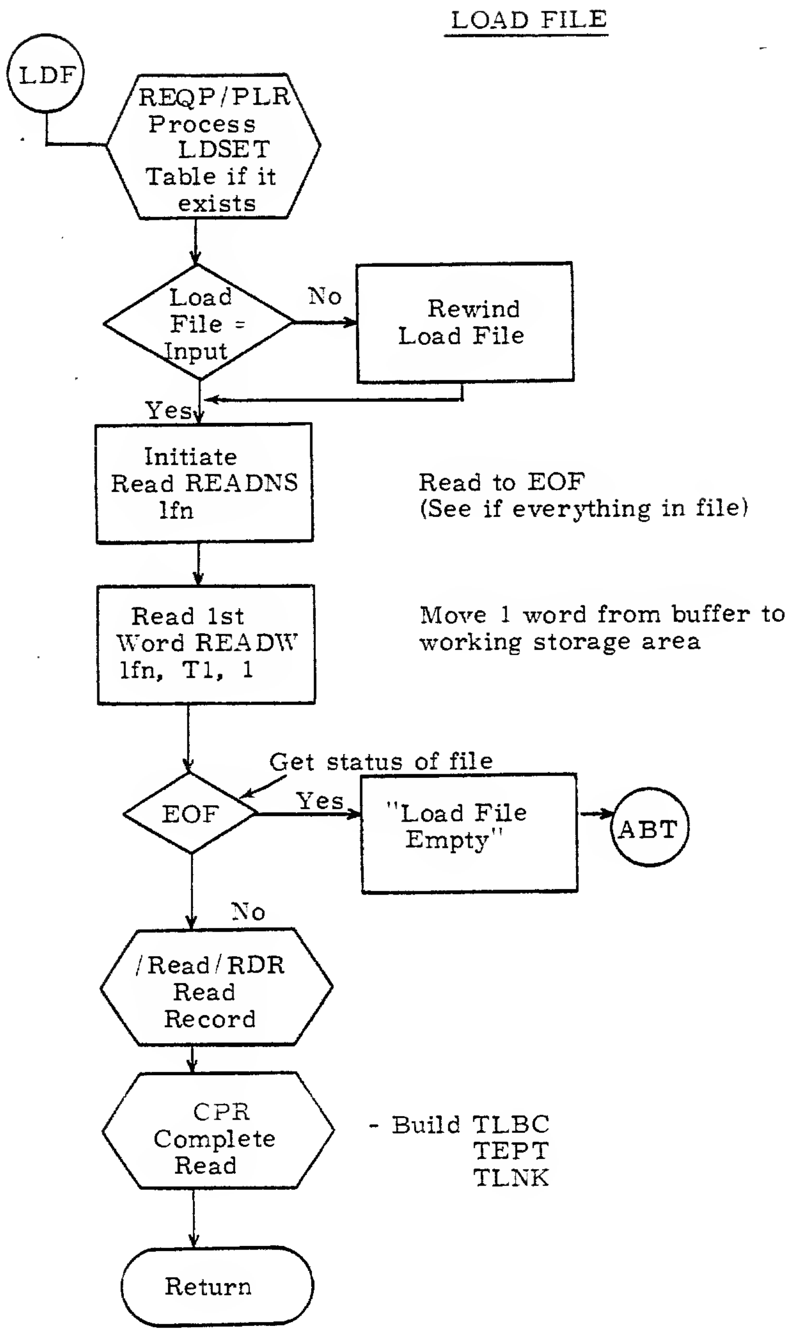


Figure 12-7. LINK Flow in Level-1 (Continued)

READ RECORD

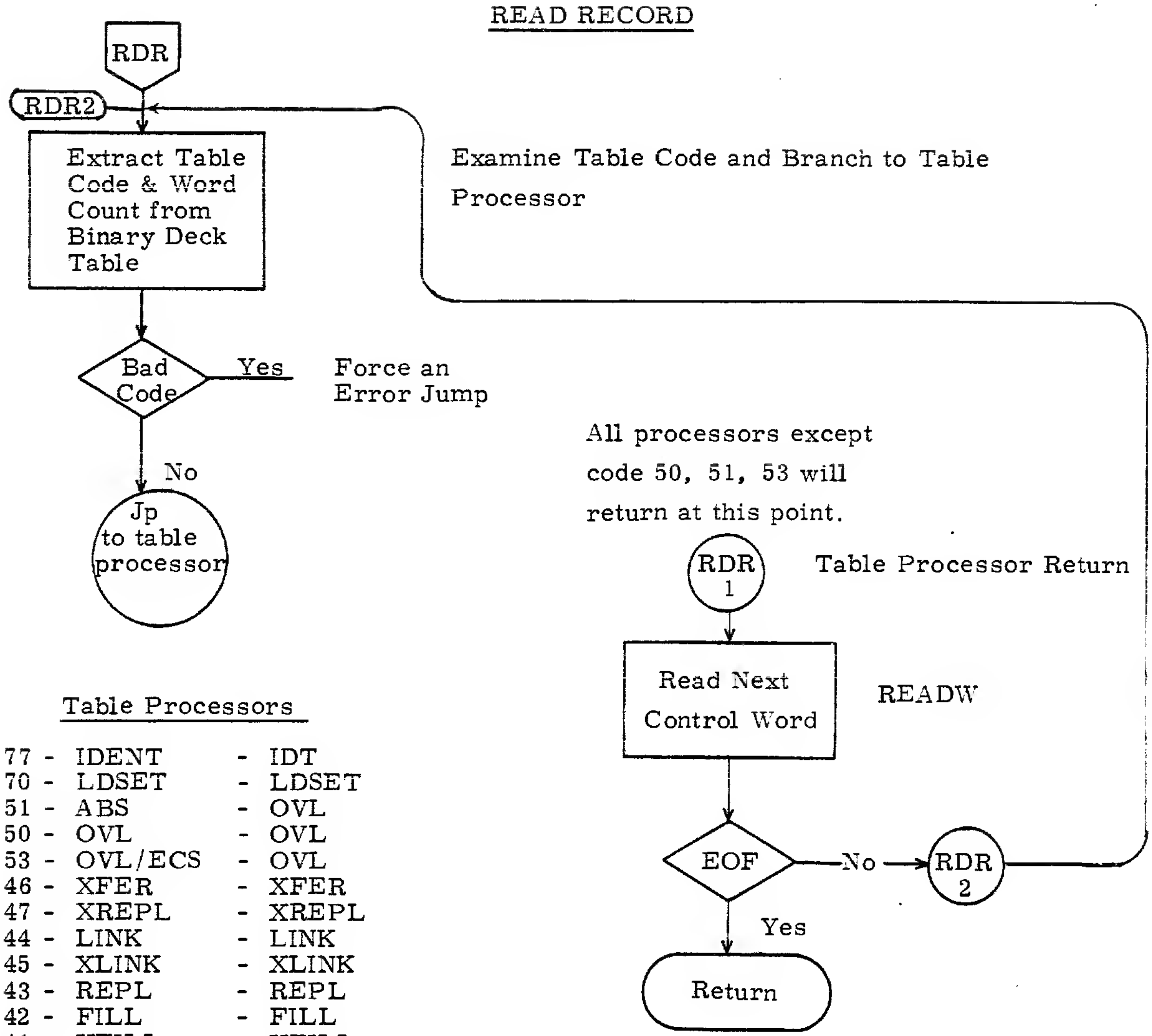


Table Processors

77 -	IDENT	- IDT
70 -	LDSET	- LDSET
51 -	ABS	- OVL
50 -	OVL	- OVL
53 -	OVL/ECS	- OVL
46 -	XFER	- XFER
47 -	XREPL	- XREPL
44 -	LINK	- LINK
45 -	XLINK	- XLINK
43 -	REPL	- REPL
42 -	FILL	- FILL
41 -	XFILL	- XFILL
40 -	TEXT	- TEXT
37 -	XTEXT	- XTEXT
36 -	ENTR	- ENTR
34 -	PIDL	- PIDL

Any Other - CKD - Loader Directive for Unidentified Table Header

Note

OVL will make RA+1 request for LDR to make an absolute load.

Figure 12-7. LINK Flow in Level-1 (Continued)

12.3 ALTERNATE SYSTEM RESIDENCY (ASR)

This feature allows moving selected portions of the operating system to non-system mass storage. Only program types OVL, ABS, and PP can be used. The program(s) are placed on the specified device after stripping off the 77 table. Accordingly, the PLD/CLD (Peripheral Library Directory/Central Library Directory) is set up to assure use of the alternate copy of a routine.

A primary area for use of this capability is high access routines. This places them on a device which has access/transfer rates superior to that of the system device. In addition, if ECS/DDP is available, PP routines which would normally be CM resident could be moved, thus freeing up CM. ECS resident CPU code is loaded directly by central monitor rather than passing through 1AJ. With the DDP option, a PP routine load may progress directly from ECS into PP core and execute. This obviates the need to shuttle the code from ECS to CM to PP core.

Alternate system devices are defined at deadstart (CMRDECK time) with the entry

$$ASR = E_1, E_2, \ldots, E_n.$$

Where E_n is the equipment ordinal. This causes a flag to be set in the appropriate MST. Alternate devices must be mass storage, ECS, non-system, non-removable, and may not be equipment 0.

The LIBDECK directive to SYSEDIT for specifying routines for a particular device is

$$*AD, E, ty_1/REC_1, ty_2/REC_2, \ldots, ty_n/REC_n$$

E selects the device and may be either an equipment ordinal or device mnemonic. In the latter case, the first alternate device of this type will be used. The record type is specified by "ty_n" and "REC_n" is the record name. Only one alternate device per routine is allowed.

The alternate PP library directory resides at the beginning of the PLD. This forces PP resident to check alternate libraries first, and also provides a mechanism for quickly disabling access to them. The PLD entry pointing at the SYSTEM copy is not removed. A pointer to the start of the SYSTEM PLD is maintained in low CM. Alternate device CPU routines are flagged by placing the equipment number in word one of the CLD entry. Word two contains track and sector for both alternate and system copies (See Figure 2-21).

An unrecoverable error while attempting to load a PP routine from an alternate device will cause all PP accesses to revert to the system device. This is accomplished by rewriting the PLD pointer (PLDP) to point at the SYSTEM copy entries. Errors encountered while

loading a CPU program will cause access to the alternate copy of the program to be disabled.

12.4 LOADER AND LOADER TABLES

When using a product set such as FTN, COBOL, etc., the product will put all the library names it needs into the LDSET table of the binary deck, so that it is no longer necessary to put a library name in the LDCW word of the CPA.

A partial list of some product libraries follow:

Library Name	Product Set
SYSLIB	Default for system
RUN2P3	RUN 2.3
BASLIB	Basic
COBOL	COBOL/SORT 4.0
FORTRAN	FTN 4.0
SIMLIB	SIMULA
SYSIO	6RM object time routine

It is important to ensure that all libraries on the system are updated to the same PSR level, as some libraries will make reference to other libraries (FORTRAN to SYSIO). If the two libraries are at different levels, they may not be compatible.

When using an ACPM table (5300), after a load, LINK will load the LWA of the largest overlay of the program (or this one if just an ABS deck) into RA+FWAS (normal default is 100) of the F . (Figure 12-8). Figure 12-10 is an example of user libraries. The user can use this to dynamically control buffers. Buffers can be started at this address and will never need to be moved or overlayed.

<table>
<tr><td>ACPM</td><td>5300</td><td>I_1</td><td>I_2</td><td>FWAS</td><td>0</td><td>entry</td><td></td></tr>
<tr><td></td><td></td><td colspan="3">fwal = 0</td><td colspan="2">wcl = 0</td><td>No ECS words</td></tr>
<tr><td></td><td>endl = 0</td><td colspan="3">ends = LWA</td><td colspan="2">WCSO</td><td></td></tr>
<tr><td></td><td colspan="6"></td><td></td></tr>
</table>

where FWAS = Address in CM to load FWA of largest overlay

entry = Address of entry point into this largest overlay

ends = LWA of largest overlay

Figure 12-8. ACPM Table

There are three ways to defeat the automatic field reduction after a load.

1. Use the control card REDUCE (-)

2. Have an external reference to the name LOADER.
   Any external reference to LOADER in a routine will allow the CP to keep all of the FL it currently has instead of the automatic reduce to FL needed for load. To accomplish this, inlcude any code like the following.

   ```
   EXT     LOADER
   SA1     =XLOADER
   etc.
   ```

3. RFL= Only used by LDR for absolute type CP loads.

The following paragraphs describe the general format for a binary deck table. All the binary deck tables are shown in Appendix D, KRONOS 2.1 Reference Manual.

In order to be externally compatible with SCOPE versions of COMPASS and other language translators, KRONOS subscribes to the SCOPE relocatable subroutine format. Hence, the logical record of output (subroutine) consists of an indefinite number of tables. Each table in this appendix is a subdivision of a logical record.

The first word of each table identifies the table to the system. That is, it indicates the kind of information that the table contains. The format of the identification word is shown in Figure 12-9 and the parameters are listed in Table 12-1.

59	53	47	35	26	17 0
cn		words		reloc	address

Figure 12-9. Identification Word

12.5 PRINTOUT EXAMPLES

Figure 12-10 is an example of User Libraries.
Figure 12-11 is an example of a FORTRAN deck with LDSET.
Figure 12-12 is an example of XFILL, XLINK Tables.
Figure 12-13 is an example of Overlay generation with an ACPM table.
Figure 12-14 is an example of an Absolute Deck
(LDSET Table not currently generated by COMPASS).
Figure 12-15 is an example of a Relocatable Program.
(LDSET Table not currently generated by COMPASS).

TABLE 12-1. IDENTIFICATION WORD PARAMETERS

Code Number (cn)	Table	reloc	address	words
34	Program Identification and Length	not used	0	Number of words in table (not counting identification word)
36	Entry Point	not used	not used	
40	Text	reloc=0, relative to RA reloc=1, relative to program origin reloc=$3-77_8$, relative to labeled common block M, where M is in position LR-2 of LCT	load address	
42	Fill	0	0	
43	Replication	not used	not used	
44	Link	not used	0	
46	Transfer	not used	not used	
77	Prefix	not used	not used	

```
Program  TRACY                                                   COMPASS3.73130        73/08/07. 12,54,51.  PAGE 2
     1
                                        IDENT      TRACY
                                        ENTRY      TRACY1
      0                      1  TRACY1  BSSZ       1
      1  7110000005 +                   MESSAGE    MESS1,,R
  2   3  0100000000 X                   RJ         =XPAUL1
      4  0400000000 +                   EQ         TRACY1
      5  05162405220504552422   MESS1   DIS        ,/ENTERED TRACY, CALL PAUL/
     10                                 END

                       43100   STORAGE USED           12 STATEMENTS        4 SYMBOLS
                               MODEL 74 ASSEMBLY      0.061 SECONDS        7 REFERENCES

                                        IDENT      TAMMY
                                        ENTRY      TAMMY1
      0                      1  TAMMY1  BSSZ       1
      1  7110000004 +                   MESSAGE    MESS2,,R
  3   3  0400000000 +                   EQ         TAMMY1
      4  05162405220504552401   MESS2   DIS        ,/ENTERED TAMMY, RETURN BONNIE/
                                        END

                       43100   STORAGE USED           11 STATEMENTS        3 SYMBOLS
                               MODEL 74 ASSEMBLY      0.059 SECONDS        6 REFERENCES

                                        IDENT      TRINA
                                        ENTRY      TRINA1
      0                      1  TRINA1  BSSZ       1
      1  7110000004 +                   MESSAGE    MESS3,,R
  4   3  0400000000 +                   EQ         TRINA1
      4  05162405220504552422   MESS3   DIS        ,/ENTERED TRINA, RETURN BONNIE/
      7                                 END

                       43100   STORAGE USED           11 STATEMENTS        3 SYMBOLS
                               MODEL 74 ASSEMBLY      0.059 SECONDS        6 REFERENCES

                                        IDENT      PAUL
                                        ENTRY      PAUL1
      0                      1  PAUL1   BSSZ       1
      1  7110000007 +                   MESSAGE    MESS4,,R
  5   3  0100000000 X                   RJ         =XKARI1
      4  7110000012 +                   MESSAGE    MESS44,,R
      6  0400000000 +                   EQ         PAUL1
      7  05162405220504552001   MESS4   DIS        ,/ENTERED PAUL, CALL KARI/
     12  22052425221655200125   MESS44  DIS        ,/RETURN PAUL, RETURN TRACY/
     15                                 END
```

USER LIBRARY: DDTLIB

Tracy calls
Paul calls
Kari
end

Tracy on PRU2
calls Paul on
PRU5 which
calls Kari on
PRU6

Figure 12-10. User Libraries

```
Program TRACY                                                          COMPASS 3:73130          73/08/07. 12,54,51   PAGE 2
                              43100  STORAGE USED                       10 STATEMENTS           5 SYMBOLS
                                     MODEL 74 ASSEMBLY                 0.075 SECONDS          10 REFERENCES
                                                  IDENT        KARI
                                                  ENTRY        KARI1
        0                         1  KARI1        BSSZ         1
    6   1  7110000004 +                           MESSAGE      MESS5,,R
        3  0400000000 +                           EQ           KARI1
        4  05162405225513012211  MESS5            DIS          ,/ENTER KARI, RETURN PAUL/
        7                                         END
                              43100  STORAGE USED                       11 STATEMENTS           3 SYMBOLS
                                     MODEL 74 ASSEMBLY                 0.059 SECONDS           6 REFERENCES
                                                  IDENT        TERRY
                                                  ENTRY        TERRY1
        0                         1  TERRY1       BSSZ         1
        1  7110000004 +                           MESSAGE      M1,,R
    7   3  0400000000 +                           EQ           TERRY1
        4  05162405220504552405  M1               DIS          ,/ENTERED TERRY, RETURN TO CALLER/
       10                                         END
                              43100  STORAGE USED                       11 STATEMENTS           3 SYMBOLS
                                     MODEL 74 ASSEMBLY                 0.034 SECONDS           6 REFERENCES


                                     JOBOAIG.  73/08/07,BAR ILAN UNIVERSITY.

                              12,54,50,JOB,T7777,OM60000.
                              12,54,50,ACCOUNT(YP)
                              12,54,50,COMPASS(B=BINARY)
                              12,54,51,  ASSEMBLY COMPLETE.  44000B SCM USED.
                              12,54,51,      0.442     CPU  SECONDS  ASSEMBLY TIME.
                              12,54,51,DEFINE(DDTLIB)
                              12,54,52,LIBGEN(F=BINARY,N=DDTLIB,P=DDTLIB,NX=0)
                              12,54,52,LIBRARY GENERATION COMPLETE.
                              12,54,52,REWIND(T=DDTLIB)
                              12,54,52,TDUMP(T=DDTLIB,0)
                              12,54,53.CP          0.532 SEC.
                              12,54,53.CM          0.003 KHH.
                              12,54,53.MS          0.077 KPR.
                              12,55,18,LP          0.300 KLN.
```

Generates a cross reference list between all programs.

Figure 12-10. User Libraries (continued)

TDUMP of USER LIBRARY

A more readable format starts on p. 12-23.1.

```
F 1 R  1 W   0- 7700 0016 0000 0000 0000   0404 2414 1102 0000 0000   5542 3650 3344 5033 4257   0000 0000 0000 0000 0000
F 1 R  1 W   4- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
-- ABOVE LINE REPEATED --                                                                            ULIB
F 1 R  1 W  14- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   7600 0000 0000 0000 0000
F 1 R  1 W  20- 2422 0103 3134 0000 0000   4000 0000 0000 0000 0002  *4000 0000 0000 0000 0005  *4000 0000 0000 0000 0006
F 1 R  1 W  24- 2401 1515 3134 0000 0000   4000 0000 0000 0000 0003   2422 1116 0134 0000 0000   4000 0000 0000 0000 0000
F 1 R  1 W  30- 2001 2514 3400 0000 0000   4000 0000 0000 0000 0005  *4000 0000 0000 0000 0006   1301 2211 3400 0000 0000
F 1 R  1 W  34- 4000 0000 0000 0000 0006   2405 2222 3134 0000 0000   4000 0000 0000 0000 0007
                                                                                                   * => EXTERNAL
-- END OF RECORD --                                                                                        REFER

F 1 R  2 W   0- 7700 0016 0000 0000 0000   2422 0103 3100 0000 0000   4236 5033 4450 3342 5755   3435 5740 3757 4034 5755
F 1 R  2 W   4- 2303 1720 0555 3557 3355   0317 1520 0123 2355 3657   4236 3436 3355 5555 5555   5555 5555 5555 5555 5555
F 1 R  2 W  10- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F 1 R  2 W  14- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   3400 0001 0000 0000 0000
F 1 R  2 W  20- 2422 0103 3100 0000 0010   3600 0002 0000 0000 0000   2422 0103 3134 0000 0000   0000 0000 0000 0100 0000
F 1 R  2 W  24- 4000 0010 0000 0100 0001   4000 4000 0000 0000 0000   7110 0000 0571 6020 0000   0100 0000 0061 0004 6000
F 1 R  2 W  30- 0100 0000 0061 0004 6000   0400 0000 0061 0004 6000   0516 2405 2205 0455 2422   0103 3156 5503 0114 1455
F 1 R  2 W  34- 2001 2514 0000 0000 0000   4000 0002 0000 0100 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F 1 R  2 W  40- 4400 0004 0000 0000 0000   2001 2514 3400 0000 0000   6001 0000 0300 0000 0000   1523 0754 0000 0000 0000
F 1 R  2 W  44- 6001 0000 0200 0000 0000
-- END OF RECORD --

F 1 R  3 W   0- 7700 0016 0000 0000 0000   2401 1515 3100 0000 0000   4236 5033 4450 3342 5755   3435 5740 3757 4034 5755
F 1 R  3 W   4- 2303 1720 0555 3557 3355   0317 1520 0123 2355 3657   4236 3436 3355 5555 5555   5555 5555 5555 5555 5555
F 1 R  3 W  10- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F 1 R  3 W  14- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   3400 0001 0000 0000 0000
F 1 R  3 W  20- 2401 1515 3100 0000 0007   3600 0002 0000 0000 0000   2401 1515 3134 0000 0000   0000 0000 0000 0100 0000
F 1 R  3 W  24- 4000 0007 0000 0100 0001   4010 0000 0000 0000 0000   7110 0000 0471 6020 0000   0100 0000 0051 0004 5000
F 1 R  3 W  30- 0400 0000 0061 0004 6000   0516 2405 2205 0455 2401   1515 3156 5522 0524 2522   1655 0217 1616 1105 0000
F 1 R  3 W  34- 4000 0002 0000 0100 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   4400 0002 0000 0000 0000
F 1 R  3 W  40- 1523 0754 0000 0000 0000   6001 0000 0200 0000 0000
-- END OF RECORD --

F 1 R  4 W   0- 7700 0016 0000 0000 0000   2422 1116 0100 0000 0000   4236 5033 4450 3342 5755   3435 5740 3757 4034 5755
F 1 R  4 W   4- 2303 1720 0555 3557 3355   0317 1520 0123 2355 3657   4236 3436 3355 5555 5555   5555 5555 5555 5555 5555
F 1 R  4 W  10- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F 1 R  4 W  14- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   3400 0001 0000 0000 0000
F 1 R  4 W  20- 2422 1116 0100 0000 0007   3600 0002 0000 0000 0000   2422 1116 0134 0000 0000   0000 0000 0000 0100 0000
F 1 R  4 W  24- 4000 0007 0000 0100 0001   4010 0000 0000 0000 0000   7110 0000 0471 6020 0000   0100 0000 0051 0004 6000
F 1 R  4 W  30- 0400 0000 0061 0004 6000   0516 2405 2205 0455 2422   1116 0156 5522 0524 2522   1655 0217 1616 1105 0000
F 1 R  4 W  34- 4000 0002 0000 0100 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   4400 0002 0000 0000 0000
F 1 R  4 W  40- 1523 0754 0000 0000 0000   6001 0000 0800 0000 0000
-- END OF RECORD --
```

Figure 12-10. User Libraries (continued)

```
F  1  R  5 W   0- 7700 0016 0000 0000 0000  2001 2514 0000 0000 0000  4236 5033 4450 3342 5755  3435 5740 3757 4074 5755
F  1  R  5 W   4- 2303 1720 0555 3557 3355  0317 1520 0123 2355 3657  4236 3436 3355 5555 5555  5555 5555 5555 5555 5555
F  1  R  5 W  10- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
F  1  R  5 W  14- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  3400 0001 0000 0000 0000
F  1  R  5 W  20- 2001 2514 0000 0000 0015  3600 0002 0000 0000 0000  2001 2514 3400 0000 0000  0000 0000 0000 0100 0000
F  1  R  5 W  24- 4000 0015 0000 0100 0001  4000 4010 0000 0000 0000  7110 0000 0771 6020 0000  0100 0000 0061 0004 6000
F  1  R  5 W  30- 0100 0000 0061 0004 6000  7110 0000 1271 6020 0000  0100 0000 0061 0004 6000  0400 0000 0061 0004 6000
F  1  R  5 W  34- 0616 2405 2205 0455 2001  2514 5603 0114 1455 1301  2211 0000 0000 0000 0000  2205 2425 2216 5520 0125
F  1  R  5 W  40- 1456 5522 0524 2522 1655  2422 0103 3100 0000 0000  4000 0002 0000 0100 0000  0000 0000 0000 0000 0000
F  1  R  5 W  44- 0000 0000 0000 0000 0000  4400 0004 0000 0000 0000  1301 2211 3400 0000 0000  6001 0000 0300 0000 0000
F  1  R  5 W  50- 1523 2754 0000 0000 0000  6001 0000 0260 0100 0005
-- END OF RECORD --

F  1  R  6 W   0- 7700 0016 0000 0000 0000  1301 2211 0000 0000 0000  4236 5033 4450 3342 5755  3435 5740 3757 4034 5755
F  1  R  6 W   4- 2303 1720 0555 3557 3355  0317 1520 0123 2355 2657  4236 3536 3355 5555 5555  5555 5555 5555 5555 5555
F  1  R  6 W  10- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
F  1  R  6 W  14- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  3400 0001 0000 0000 0000
F  1  R  6 W  20- 2001 2514 0000 0000 0015  3600 0002 0000 0000 0000  1301 2211 3400 0000 0000  0000 0000 0000 0100 0000
F  1  R  6 W  24- 4000 0007 0000 0100 0001  4010 0000 0000 0000 0000  7110 0000 0471 6020 0000  0100 0000 0061 0004 6000
F  1  R  6 W  30- 0400 0000 0061 0004 6000  0516 2405 2255 1301 2211  5655 2205 2425 2216 5520  0125 1400 0000 0000 0000
F  1  R  6 W  34- 4000 0002 0000 0100 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  4400 0002 0000 0000 0000
F  1  R  6 W  40- 1523 0754 0000 0000 0000  6001 0000 0200 0000 0000
-- END OF RECORD --

F  1  R  7 W   0- 7700 0016 0000 0000 0000  2405 2222 3100 0000 0000  4236 5033 4450 3342 5755  3435 5740 3757 4034 5755
F  1  R  7 W   4- 2303 1720 0555 3557 3355  0317 1520 0123 2355 3657  4236 3436 3355 5555 5555  5555 5555 5555 5555 5555
F  1  R  7 W  10- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
F  1  R  7 W  14- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  3400 0001 0000 0000 0000
F  1  R  7 W  20- 2405 2222 3100 0000 0010  3600 0002 0000 0000 0000  2405 2222 3134 0000 0000  0000 0000 0000 0100 0000
F  1  R  7 W  24- 4000 0010 0000 0100 0001  4010 0000 0000 0000 0000  7110 0000 0471 6020 0000  0100 0000 0061 0004 6000
F  1  R  7 W  30- 0400 0000 0061 0004 6000  0516 2405 2205 0455 2405  2222 3156 5522 0524 2522  1655 2417 5503 0114 1405
F  1  R  7 W  34- 2200 0000 0000 0000 0000  4000 0002 0000 0100 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
F  1  R  7 W  40- 4400 0002 0000 0000 0000  1523 1754 0000 0000 0000  6001 0000 0200 0000 0000
-- END OF RECORD --

F  1  R 10 W   0- 7700 0016 0000 0000 0000  0404 2414 1102 0000 0000  5542 3630 3344 5033 4257  0000 0000 0000 0000 0000
F  1  R 10 W   4- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
-- ABOVE LINE REPEATED --                                                                          INDEX
F  1  R 10 W  14- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  7000 0000 0000 0000 0016
F  1  R 10 W  20- 0404 2414 1102 0000 0005  0000 0000 0000 0000 0001  2422 0103 3100 0000 0003  0000 0000 0000 0000 0002
F  1  R 10 W  24- 2401 1515 3100 0000 0003  0000 0000 0000 0000 0003  2422 1116 0100 0000 0003  0000 0000 0000 0000 0004
F  1  R 10 W  30- 2001 2514 0000 0000 0003  0000 0000 0000 0000 0005  1301 2211 0000 0000 0003  0000 0000 0000 0000 0006
F  1  R 10 W  34- 2405 2222 3100 0000 0003  0000 0000 0000 0000 0007
-- END OF RECORD --
-- END OF FILE --
-- END OF INFORMATION --
-- END OF DUMP --
```

Figure 12-10. User Libraries (continued)

This is a copy of the dump on p. 12-22 and 12-23 in a more readable format.

USER LIBRARY EXAMPLE

DDTLIB

Record 1

77000016000000000000	LABEL HEADER
04042414110200000000	DDTLIB
55423650334450304257	73/09/07
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
76000000000000000000	ULIB HEADER
24220103313400000000	TRACY1
40000000000000000002	code in record 2
* 40000000000000000005	extern reference to PAUL 1 in rec 5
* 40000000000000000006	extern ref to KARI1 from PAUL1 in rec 6
24011515313400000000	TAMMY1
40000000000000000003	code in rec 3
24221116013400000000	TRINA1
40000000000000000004	code in rec 4
20012514340000000000	PAUL1
40000000000000000005	code in rec 5 compare to TRACY1 extern
* 40000000000000000006	extern ref to KARI1 in rec 6
13012211340000000000	KARI1
40000000000000000006	code in rec 6
24052222313400000000	TERRY1
40000000000000000007	code in rec 7

*indicates external reference

USER LIBRARY STRUCTURE

Record 2

```
       77000016000000000000
       24220103310000000000
       33365035355042355547
       33335733425734405755
       23031720055536573755
       03171520012323553657
       42343543365555555555
       55555555555555555555
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       --------------------
TRACY  34000001000000000000
       24220103310000000010
       36000002000000000000
       24220103313400000000
       00000000000001000000
       40000010000001000001
       10004000000000000000
       43601206277110000005
       01000000006100046000
       01000000006100046000
       04000000006100046000
       05162405220504552422
       01033156550301141455
       20012514000000000000
       40000002000001000000
       00000000000000000000
       00000000000000000000
       44000004000000000000
       20012514340000000000
       60010000030000000000
       15230754000000000000
       60010000020000000000
```

Record 3

```
       77000016000000000000
       24011515310000000000
       33365053535504235547
       33335733425734405755
       23031720055536573755
       03171520012323553657
       42343543365555555555
       55555555555555555555
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       00000000000000000000
       --------------------
TAMMY  34000001000000000000
       24011515310000000007
       36000002000000000000
       24011515313400000000
       00000000000001000000
       40000007000001000001
       10100000000000000000
       43601206277110000004
       01000000006100046000
       04000000006100046000
       05162405220504552401
       15162405220504552401
       16550217161611050000
       40000002000001000000
       00000000000000000000
       00000000000000000000
       44000002000000000000
       15230754000000000000
       60010000020000000000
```

Record 4

```
      77000016000000000000
      24222111601000000000
      33365035355042355547
      33335733425734405744
      23031720055536573755
      03171520012323553657
      42343543365555555555
      55555555555555555555
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      34000001000000000000
TRINA 24222111601000000007
      36000002000000000000
      24222111601340000000
      00000000000001000000
      40000007000001000001
      10100000000000000000
      43601206277110000004
      01000000006100046000
      04000000006100046000
      05162405220504552422
      11160156552205242522
      16550217161611050000
      40000002000001000000
      00000000000000000000
      00000000000000000000
      44000002000000000000
      15230754000000000000
      60010000020000000000
```

Record 5

```
     77000016000000000000
     20012514000000000000
     33365035355042355547
     33335733425734415755
     23031720055536573755
     03171520012323553657
     42343543365555555555
     55555555555555555555
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     34000001000000000000
PAUL 20012514000000000015
     36000002000000000000
     20012514340000000000
     00000000000001000000
     40000015000001000001
     10001010000000000000
     43601206277110000007
     01000000006100046000
     01000000006100046000
     43601206277110000012
     01000000006100046000
     04000000006100046000
     05162405220504552001
     25162405220504552001
     22110000000000000000
     22052425221655200125
     14565522052425221655
     24220103310000000000
     40000002000001000000
     00000000000000000000
     00000000000000000000
     44000004000000000000
     13012211340000000000
     60010000030000000000
     15230754000000000000
     60010000026001000005
```

Record 6

```
     77000016000000000000
     13012211000000000000
     33365035355042355547
     33335733425734425755
     23031720055536573755
     03171520012323553657
     42343543365555555555
     55555555555555555555
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     34000001000000000000
KARI 13012211000000000007
     36000002000000000000
     13012211340000000000
     00000000000001000000
     40000007000001000001
     10100000000000000000
     43601206277110000004
     01000000006100046000
     04000000006100046000
     05162405225513012211
     56552205242522165520
     01251400000000000000
     40000002000001000000
     00000000000000000000
     00000000000000000000
     44000002000000000000
     15230754000000000000
     60010000020000000000
```

Record 7

```
      77000016000000000000
      24052222310000000000
      33365035355042355547
      33335733425734425755
      23031720055536573755
      03171520012323553657
      42343543365555555555
      55555555555555555555
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      00000000000000000000
      34000001000000000000
TERRY 24052222310000000010
      36000002000000000000
      24052222313400000000
      00000000000001000000
      40000010000001000001
      10100000000000000000
      43601206277110000004
      01000000006100046000
      04000000006100046000
      05162405220504552405
      22223156552205242522
      16552417550301141405
      22000000000000000000
      20000001000002000000
      00000000000000000000
      00000000000000000000
      44000002000000000000
      15230754000000000000
      60010000020000000000
```

Record 10

77000016000000000000	LABEL HEADER
04042414110200000000	DDTLIB
55423630334450334257	73/09/07
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
00000000000000000000	
70000000000000000016	INDEX HEADER
04042414110200000005	DDTLIB
00000000000000000001	library in rec 1
24220103310000000003	TRACY
00000000000000000002	code in rec 2
24011515310000000003	TAMMY
00000000000000000003	code in rec 3
24221116010000000003	TRINA
00000000000000000004	code in rec 4
20012514000000000003	PAUL
00000000000000000005	code in rec 5
13012211000000000003	KARI
00000000000000000006	code in rec 6
24052222310000000003	TERRY
00000000000000000007	code in rec 7

```
                                                                      COMPASS 3.73130          73/08/07. 12,54,51  PAGE2
                                                  IDENT          TRACY
                                                  ENTRY          TRACY1
  0  0000000000                     TRACY1        PS
  1  7110000007  +                                MESSAGE        T1,,R
  3  7110000012  +                                MESSAGE        T2,,R
  5  0100000000  +                                RJ             =XPAUL1
  6  0400000000  +                                EQ             TRACY1
  7  47474723012411230611  T1                     DIS            ,/***SATISFIED FROM ALTLIB***/
 12  47474705162405225524  T2                     DIS            ,/***ENTER TRACY, CALL PAUL***/
 15                                               END
                          43100  STORAGE USED                             18 STATEMENTS          5 SYMBOLS
                                 MODEL 74 ASSEMBLY                      0.083 SEONDS            10 REFERENCES

                                                  IDENT          REMARKS
                                                  ENTRY          REMARK
  0  0000000000                     REMARK        PS
  1  7110000004  +                                MESSAGE        M1,,P
  3  0400000000  +                                EQ             REMARK
  4  53535323012411230611  M1                     DIS            ,/$$$SATISFIED FROM ALTLIB$$$/
  7                                               END
                          41300  STORAGE USED                             11 STATEMENTS          3 SYMBOLS
                                 MODEL 74 ASSEMBLY                      0.033 SECONDS            6 REFERENCES

                   TRACAIJ.  73/09/07.RAR ILAN UNIVERSITY.
        12,59,40.TRACY,T100,CM50000.                                                          USER LIBRARY: ALTLIB
        12,59,40.ACCOUNT,VR)
        12,59,40.COMPASS.
        12,59,40.ASSEMBLY COMPLETE.   44000B SCM USED.
        12,59,40,    0.161     CPU SECONDS    ASSEMBLY TIME?
        12,59,41.DEFINE(ALTLIB)
        12,59,41.LIBGEN(N=ALTLIB,P=ALTLIB)
        12,59,41.LIBRARY GENERATION COMPLETE.
        12,59,41.CP          0.189 SEC.
        12,59,41.CM          0.001KWH
        12,59,41.MS          0.020 KPR.
        12,59,41.LP          0.126 KLN.
```

Figure 12-10. User Libraries (continued)

```
                                                         COMPASS 3.73130          73/09/07. 13.45.14 PAGE 2
                                     IDENT            CHECK
                                     ENTRY            ST
 0 7110000007 +          ST          MESSAGE          M1,,R
 2 0100000000 X                      RJ               =XTRACY1
 3 7110000013 +                      MESSAGE          M2,,R
 5 7160247021                        ENDRUN
 7 06162405225503100503  M1          DIS              ,/ENTER CHECK,CALL EXTERNAL REFERENCE/
13 22052425221655031005  M2          DIS              ,/RETURN CHECK, TERMINATE/
16                                   END              ST

                 43100   STORAGE USED                      20 STATEMENTS          6 SYMBOLS
                         MODEL 74 ASSEMBLY               0.064 SECONDS           10 REFERENCES
```

CHECK
SYMBOLIC REFERENCE TABLE.

MSG=	0	EXTERNAL=	2/04	2/06
M1	7	PROGRAM=	2/03	2/07 L
M2	13	PROGRAM=	2/05	2/08 L
ST	0	PROGRAM=	2/02 E	2/03 L
SYS	0	EXTERNAL=	2/07	
TRACY1	0	EXTERNAL=	2/04	

LOAD MAP.
BLOCK ASSIGNMENTS. 1

LIBRARY

BLOCK	ADDRESS	LENGTH	FILE
CHECK	100	16	LGO
TRACY	116	10	DDTLIB
PAUL	126	15	DDTLIB
KARI	143	7	DDTLIB
CPUSYS	152	32	SYSLIB

LOAD MAP.
BLOCK ASSIGNMENTS. 2

LDSET (USE=)

BLOCK	ADDRESS	LENGTH	FILE
CHECK	100	16	LGO
TRACY	116	15	ALTLIB ✓
PAUL	133	15	DDTLIB
KARI	150	7	DDTLIB
CPUSYS	157	32	SYSLIB

LOAD MAP.
BLOCK ASSIGNMENTS. 3

BLOCK	ADDRESS	LENGTH	FILE
CHECK	100	16	LGO
TRACY	116	10	DDTLIB
TAMMY	126	7	DDTLIB
TRINA	135	7	DDTLIB
PAUL	144	15	DDTLIB
KARI	161	7	DDTLIB
CPUSYS	170	32	SYSLIB

LDSET (USEP=)
These routines are forced to be loaded even though never used.

LOAD MAP
BLOCK ASSIGNMENTS 4

BLOCK	ADDRESS	LENGTH	FILE
CHECK	100	16	LGO
TRACY	116	10	DDTLIB
PAUL	126	15	DDTLIB
KARI	143	7	DDTLIB
TERRY	152	10	DDTLIB
CPUSYS	162	32	SYSLIB

Figure 12-10. User Libraries (continued)

```
13, 45, 14, 001, 0M50000.
13, 45, 14, ACCOUNT, YP)
13, 45, COMPASS.
13, 45, 15. ASSEMBLY COMPLETE.  44000B SCM USED.
13, 45, 15       0.120  CPU SECONDS  ASSEMBLY TIME.
13, 45, 15. ATTACH(DDTLIB. ALTLIB)
13, 45, 15, LIBRARY(DDTLIB)
13, 45, 15. LDSET(MAP=B)                  ) 1
13, 45, 15. LGO.
13, 45, 15, ENTER CHECK, CALL EXTERNAL REFERENCE
13, 45, 15, ENTERED TRACY,  CALL PAUL
13, 45, 15. ENTERED PAUL, CALL KARI
13, 45, 15. ENTER KARI,  RETURN PAUL
13, 45, 15. RETURN PAUL,  RETURN TRACY
13, 45, 15. RETURN CHECK, TERMINATE
13, 45, 15.*
13, 45, 16.*
13, 45, 16. LIBRARY(DDTLIB)
13, 45, 16. LDSET(MAP=B)                          ) 2
13, 45, 16. LDSET(LIB=ALTLIB, MAP=B)
13, 45, 16LGO.
13, 45, 16. ENTER CHECK, CALL EXTERNAL REFERENCE
13, 45, 16*** SATISFIED FROM ALTLIB***
13, 45, 16. ***ENTER TRACY,  CALL PAUL***
13, 45, 16, ENTERED PAUL, CALL KARI
13, 45, 16ENTER KARI,  RETURN PAUL
13, 45, 16. RETURN PAUL,  RETURN TRACY
13, 45, 16. RETURN CHECK, TERMINATE
13, 45, 16.*
13, 45, 16.*
13, 45, 16. LIBRARY(DDTLIB)                       Forced Loading of Tammy/Trina
13, 45, 16. LDSET(MAP=B)                  ) 3     even though check does not
13, 45, 16. LDSET(USE=TAMMY/TRINA)                reference the programs
13, 45, 16. LGO.
13, 45, 17. ENTER CHECK, CALL EXTERNAL REFERENCE
13, 45, 17. ENTERED TRACY,  CALL PAUL
13, 45, 17. ENTERED PAUL, CALL KARI
13, 45, 17. ENTER KARI,  RETURN PAUL
13, 45, 17. RETURN PAUL,  RETURN TRACY
13, 45, 17. RETURN CHECK,  TERMINATE
13, 45, 17.*
13, 45, 17*
13, 45, 17. LDSET(USE=PAUL1/KARI/TERY1)   ) 4   These are entry-points
13, 45, 17. LDSET(MAP=B)                        which force the routines
13, 45, 17. LGO.                                   with these entry-points
13, 45, 18. ENTER CHECK, CALL EXTERNAL REFERENCE          to load
13, 45, 18. ENTERED TRACY,  CALL PAUL
13, 45, 18. ENTERED PAUL,  CALL KARI
13, 45, 18. ENTER KARI,  RETURN PAUL
13, 45, 18. RETURN PAUL,  RETURN TRACY
13, 45, 18. RETURN CHECK,  TERMINATE
13, 45, 18.*                          Force no-load of routine Tracy
13, 45, 18.*
13, 45, 18. LDSET(OMIT=TRACY)     ) 5
13, 45, 18. LGO.
13, 45, 18. ENTER CHECK, CALL EXTERNAL REFERENCE
13, 45, 18. ARITH. ERROR 1 AT 400103.    Note Tracy not loaded and the call to
13, 45, 18. CP     0.389 SEC.              Tracy1 makes an unsatisfied external
13, 45, 18. CM     0.002 KWH.            A call to Tracy1 initiates message above.
13, 45, 18. MS     0.145 KPR.
```

Figure 12-10. User Libraries (continued)

```
PROGRAM CHK(OUTPUT, TAPE1, TAPE2)
DIMENSION IBUFF(128)
10 FORMAT 13I10)
DO 100 N=10, 120, 20
DO 50 I=1, N
50 IBUFF(I)  N
WRITE (1) (IBUFF(M), M=1,N)
WRITE (2,10) (IBUFF(MM), MM=1,N)
100 CONTINUE
STOP
END
```

FTN produced Binary Deck

```
        77000016000000000000
        03101300000000000000   CHK
        33405033365042355555
        33345744425733405747
        23031720055536573755   SCOPE 3.4
        06?41655555555375733   FTN 4.0
        42353335424141414155
        55555555555555555555
        55555555555555555555
        55555555555555555555
        55555555555555555555
        55555555555555555555
        55555555555555555555
        55555555555555555555
        55555555555555555555
LDSET   70000008000000000000
LIS     00100002000000000000             } LDSET
        06172224220116000000   FORTRAN   } Table
        23312311170000000000   SYSIO     }
        34000001000000000000
PIDL    03101300000000006407
        36000010000000000000
ENTER   03101300000000000000
        00000000000001006137
        17252420252460000000
        00000000000001000000
        24012005346000000000
        00000000000001002036
        24012006356000000000
        00000000000001004074
        40000010000001000000
TEXT    00400002000000000000
        00000000000000000000
        00000000000000000022
        00000000406000000000
        00000000002000000000
        00000000000000000000
        00000000000000000034
        00000000000000000000
        43000002000000000001
REPL    00000000000001000006
        00002500000000000000
        40000010000001002036
        00400002000000000000
        00000000000000000000
        00000000000000002060
        00000000406000000000
```

```
        00000000000001004102
REPL    00000000000000000000
        40000000000101006137
TEXT    40506000500810002202
        51100061320100000000
        71700000125170006203
        51500062036120000001
        62780062066130006203
        61100062074400046000
        56530107555671066121
        06710061445150006203
        71000000010160000002
        37450273007100006207
        10244206605110006172
        21273120062053015642
        72460000013670627004
        42610517000617426706
        51700061734600046000
        01000000000007006136
        40000006100001006132
        10420040000000000000
TEXT    17252425252000000000
        24012005340000002036
        24012005350000004074
        77777777770000000000
        03101355555555006137
        40000010000001008155
TEXT    41104050004040000000
        51500047037100000002
        20060717000620712670
        51100161762053036765
        51700062004600046000
        01000000000010006136
        51500062037276000024
        72077776065075046000
        03300061415110006202
        04000000004000046000
        51100051360400000000
        40000014000001006170
        00020000104000000000
TEXT    55343355555500000000
        51343611343352555565
        00000000000000002036
        77777777777777777776
        77777777777777777776
        00000000000000000000
        00000000000000004074
```

Figure 12-11. FORTRAN Deck with LDSET.

```
      00000000002000000000          00000000000000006170
      00000000000000000000          77777777777777777776
TEXT  00000000000000002072    TEXT  00000000000000000000
      00000000000000000000          00000000000000000000
      ____________________          ____________________
      43000002000000000001          44000010000000000000
REPL  00000000000001002044          05160457000000000000
      00002500000000000000          40010061672324172057
      ____________________          00000000006001006166
      40000010000001004074    LINK  17252403115700000000
      00400002000000000000          60010061621725240211
TEXT  00000000000000000000          57000000006001006155
      00000000000000004116          21431624223157000000
      00000000406000000000          40010061370000000000
      00000000002000000000          ____________________
      00000000000000000000    XFER  46000001000000000000
                                    03101300000000000000
```

Figure 12-11. FORTRAN Deck with LDSET. (continued)

```
                                    IDENT    TST
                                    ENTRY    ST                  XFILL/XLINK   Binary Deck
                                *                                                 Tables
                                *
0                        62     TAG      BSS      50
62                       24     A        BSS      20
106                      36     B        BSS      30
144                     144     C        BSS      100
                                *
                                *
310                             ST       BSS      0
310  0400000310 +                        EQ       ST
                                *
                                *
311  00000000 +                 XFILL    VFD      24/TAG,18/A,18/B  }
             000062 +                                               }  Generates XFILL Table
                  000106 +                                          }
312  00000000000000000144 +              VFD      60/C              }
                                *
                                *                                   }
313  00000000000000000000 X     XLINK    VFD      60/-XDUMMY        }  Generates XLINK Table
314  00000000 X                          VFD      24/=XDUMMY,36/0   }
             000000000000
                                *
                                *
315                                      END

                       41300    STORAGE USED          23 STATEMENTS      8 SYMBOLS
                                MODEL 74 ASSEMBLY     0.052 SECONDS      15 REFERENCES

SYMBOLIC REFERENCE TABLE.

A           62    PROGRAM*     2/06L    2/16
B          106    PROGRAM*     2/07L    2/17
C          144    PROGRAM*     2/08L    2/18
DUMMY        0    EXTERNAL*    2/21     2/22
ST         310    PROGRAM*     2/02E    2/11L    2/12
TAG          0    PROGRAM*     2/05L    2/15
XFILL      311    PROGRAM*     2/15L
XLINK      313    PROGRAM*     2/21L
```

Figure 12-12. XFILL, XLINK Tables.

```
         - FILE DUMP -             TDUMP(I=LGO,O)                                          73/09/07. 13,32,16   PAGE 1

F 1 R 1 W   0- 7700 0016 0000 0000 0000  2423 2400 0000 0000 0000  4236 5033 4450 3342 5755  3436 5736 3557 3441 5755
F 1 R 1 W   4- 2303 1720 0555 3557 3355  0317 1520 0123 2355 3657  4236 3436 3355 5555 5555  5555 5555 5555 5555 5555
F 1 R 1 W  10- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
F 1 R 1 W  14- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  3400 0001 0000 0000 0000
F 1 R 1 W  20- 2423 2400 0000 0000 0315  3600 0002 0000 0000 0000  2324 0000 0000 0000 0000  0000 0000 0000 0100 0310
F 1 R 1 W  24- 0000 0006 0000 0100 0310  4040 0000 0000 0000 0000  0400 0003 1061 0004 6000  0000 0000 0000 6200 0106
F 1 R 1 W  30- 0000 0000 0000 0000 0144  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  4100 0003 0000 0000 0000
F 1 R 1 W  34- 0000 0003 1122 2200 1001  0000 0003 1144 3000 1001  0000 0003 1200 7400 1001  4500 0003 0000 0000 0000
F 1 R 1 W  40- 0425 1515 3100 0000 0000  0000 0003 1300 7400 0001  0000 0003 1444 3000 0001
                                                                                            XLINK
                                                                                         XFILL
-- END OF RECORD --
-- END OF INFORMATION --
-- END OF DUMP --
```

Figure 12-12. XFILL, XLINK Tables. (continued)

1

```
      OVERLAY(XXXXXX,0,0)
      PROGRAM XTST(INPUT,OUTPUT)
      PRINT 500
500   FORMAT(*MAIN*)
      CALL OVERLAY(6HXXXXXX,
                   1,0,6HRECALL)
      CALL OVERLAY(6HXXXXXX,
                   2,0,6HRECALL)
      END
```

Forces 6RM Code of minimum size 10K octal

2

```
      OVERLAY(XXXXXX,1,0)
      PROGRAM XTST1
      PRINT 500
500   FORMAT(* OVERLAY 1*)
      END
```

3

```
      OVERLAY(XXXXXX,2,0)
      PROGRAM XTST2
      PRINT 500
500   FORMAT(* OVERLAY 2*)
      END
```

Address of entry-point into this largest overlay

1

```
  - FILE DUMP -           TDUMP(I XXXXXX,0)                          73/09/07. 11.14 10    PAGE 1
F  1 R  1 W   0- 7700 0016 0000 0000 0000  3024 2324 0000 0000 0000  5542 3650 3344 5033 4257  3024 2324 0000 0000 0000
F  1 R  1 W   4- 4236 5033 4450 3342 5755  3434 5734 3757 3342 5755  1322 1716 1723 3557 3455  0624 1655 5555 5537 5733
F  1 R  1 W  10- 4236 3440 3541 4141 3055  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555
F  1 R  1 W  14- 5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5300 0000 0001 0000 4201
F  1 R  1 W  20- 0000 0000 0000 0000 0000  0000 0000 0164 6500 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0123
F  1 R  1 W  24- 0000 0000 4060 0000 0000  0000 0000 0020 0200 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0135
F  1 R  1 W  30- 0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000  0000 0000 0000 0000 0000
```

LWA of Largest Overlay

Completion of loading, place in RA+100 LWA of largest overlay

2

```
F  1 R  2 W   0- 7700 0016 0000 0000 0000  3024 2324 3400 0000 0000  5542 3650 3344 5033 4257  3434 5734 3757 3342 5755
F  1 R  2 W   4- 1322 1716 1723 3557 3455  0624 1655 5555 5537 5733  4236 3440 3541 4141 3055  5555 5555 5555 5555 5555
F  1 R  2 W  10- 5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555
F  1 R  2 W  14- 5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5000 0100 0164 5001 6453
F  1 R  2 W  20- 7777 7777 7777 7776 6167  3024 2324 3455 5501 6453  5110 0164 5101 0000 4275  5110 0164 5746 0004 6000
F  1 R  2 W  24- 0100 0067 7200 0301 6452  5110 0164 5204 0000 4526  0000 0000 0000 0000 2137  0000 0000 0000 0001 6467
F  1 R  2 W  30- 0000 0000 0000 0000 0000  5540 3333 5555 0000 0000  5147 5517 2605 2214 0131  5534 4752 5555 5555 5555
-- END OF RECORD --
```

3

```
F  1 R  3 W   0- 7700 0016 0000 0000 0000  3024 2324 3500 0000 0000  5542 3650 3344 5033 4257  3434 5734 3757 3343 5755
F  1 R  3 W   4- 1322 1716 1723 3557 3455  0624 1655 5555 5537 5733  4236 3440 3541 4141 3055  5555 5555 5555 5555 5555
F  1 R  3 W  10- 5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555
F  1 R  3 W  14- 5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5555 5555 5555 5555 5555  5000 0200 0164 5001 6453
F  1 R  3 W  20- 7777 7777 7777 7776 6167  3024 2324 3555 5501 6453  5110 0164 5101 0000 4275  5110 0164 5746 0004 6000
F  1 R  3 W  24- 0100 0067 7200 0301 6452  5110 0164 5204 0000 4526  0000 0000 0000 0000 2137  0000 0000 0000 0001 6462
F  1 R  3 W  30- 0000 0000 0000 0000 0000  5540 3333 5555 0000 0000  5147 5517 2605 2214 0131  5585 4752 5555 5555 5555
-- END OF RECORD --

-- END OF INFORMATION --

-- END OF DUMP --
```

Figure 12-13. Overlay Generation With ACPM Table

```
                                                      IDENT      CHECK,CHECK
                                                      ABS
                                                      ENTRY      ONE
                                                      ENTRY      TWO
200                                                   ORG        200B
200                                        CHECK      BSS        0
200  7110000206                            ONE        MESSAGE    MESS,3,R
                     7160200003                SX6  200000B+3
201  0100000242                                RJ  =XMSG-
202  7110000211                            TWO        MESSAGE    MESS1,3,R
                     7160200003                SX6  200000B+3
203  0100000242                                RJ  =XMSG=
204  7160247021                                           ENDRUN
                                               SX6 4RENDP/4
                    20650                      LX6 40D
205  0100000217                                RJ  =XSYS=
206  55031005031355010223                  MESS       DIS        ,/CHECK ABSOLUTE PGM /
207  17142524055520071555
210  00000000000000000000
211  55474747474747474747                  MESS1      DIS        ,/******************** /
212  47474747474747474747
213  47000000000000000000
214                                                   XTEXT      COMCSYS
        77 Table      16 Words
     77000016000000000000
     03100503130000000000
     55423450333750334257   71/04/01
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000
     00000000000000000000   Table
     03152055355733000000   CMP 2.0
ABS  51000000000175000002
ONE  17160500000000000200
TWO  24271700000000000202
     71100002067160200003— First Word Of Object Code
     01000002426100046000
     71100002117160200003
     01000002426100046000
     71602470212065046000
     01000002176100046000
     55031005031355010223
     17142524055520071555
     00000000000000000000
     55474747474747474747
     47474747474747474747
     47000000000000000000
```

Figure 12-14. Absolute Deck

```
54110201230331000216
01300000006100046000
04000002226100046000
00000000006100046000
51100000010311000220
54610040000021646000
51100002141061146000
51600002165110000001
10611010000021546000
20652010000021746000
51100000010311000226
00000000006100046000
51100000010311000227
71602203140400000225
20150366610100000217
00000000006100046000
71602203142065236662
53160201730331000233
03010002335110000001
03110002357110000001
04000002326100046000
12661010000021746000
00000000006100046000
20630121617160152307
20652040000024146000
```

Figure 12-14. Absolute Deck (Continued)

```
                                                  IDENT         PRG1
                                                  LIST          L,R,G,D
                                                  ENTRY         FIRST,INTMULT
 0  5110000011 +                    FIRST         SA1           INTEGER
              5120000012 +                        SA2           INTEGER+1
 1  0100000005 +                                  RJ            INTMULT
 2  5170000000 C                                  SA7           PARAM
              0100000000 X                        RJ            =XSECOND
 3  7160247021                                    ENDRUN
              20650                   LX6 40D
 4  0100000000 X                      RJ =XSYS=
                                    *
 5  00000000000000000000            INTMULT       DATA          0               INTEGER MULTIPLY
 6  27101                                         PX1           BO,X1
         24101                                    NX1           BO,X1
              27202                               PX2           BO,X2
                   24202                          NX2           BO,X2
 7  40712                                         FX7           X1*X2
         26717                                    UX7           B1,X7
              22717                               LX7           B1,X7
10  0400000005 +                                  EQ            INTMULT
                                    *
11  00000000000000000024            INTEGER       DATA          20,3
12  00000000000000000003
                                                  USE           /BLK/
 0                                  PARAM         BSSZ          5
                                                  USE           //
 0                                  BLANK         BSS           10
                                                  END           FIRST
                                    DEFAULT SYMBOLS DEFINED BY COMPASS.

 0  X                               SECOND
 0  X                               SYS=

13


                                                  IDENT         PRG2
                                                  LIST          L,R,G,D
                                                  ENTRY         SECOND
 0  00000000000000000000            SECOND        DATA          0
 1  5110000000 C                                  SA1           VALUES
              10211                               BX2           X1
 2  0100000000 X                                  RJ            =XINTMULT
 3  5170000001 C                                  SA7           VALUES+1
              0400000000 +                        EQ            SECOND
                                                  USE           /BLK/
 0                                  VALUES        BSS           5

                                    DEFAULT SYMBOLS DEFINED BY COMPASS.

 0  X                               INTMULT

 4                                                END
```

Figure 12-15. Relocatable Program

PRG1

```
77000016000000000000
20220734000000000000
55423450333750334257
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
03152055355733000000   CMP 2.0
34000003000000000000
20220734000000000013
02141300000000000005
55555555555555000012
36000004000000000000
06112223240000000000
00000000000001000000
11162415251424000000
00000000000001000005
40000014000001000000
52000000001000000000
51100000115120000012
01000000056100046000
51700000000100000000
71602470212065046000
01000000006100046000
00000000000000000000
27101241012720224202
40712267172271746000
04000000056100046000
00000000000000000024
00000000000000000003
40000002000003000000
00000000000000000000
00000000000000000000
43000002000000000001
00000000000003000000
00000400000000000000
42000001000000000000
00000000036001000002
44000004000000000000
23312354000000000000
60010000040000000000
23503171604000000000
40010000020000000000
46000001000000000000
06112223240000000000
```

PRG 2

```
77000016000000000000
20220735000000000000
55423450333750334257
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
03152055355733000000   CMP 2.0
34000002000000000000
20220735000000000004
02141300000000000005
36000002000000000000
23503171604000000000
00000000000001000000
40000005000001000000
00001000000000000000
00000000000000000000
51100000001021146000
01000000000100046000
51700000010400000000
42000002000000000000
00000000036001000001
60010000030000000000
44000002000000000000
11162415251424000000
60010000020000000000
```

Figure 12-15. Relocatable Program (continued)

13.0 INTRODUCTION

TELEX is a subsystem that provides support for interactive processing from remote terminals such as TTYs (Teletypewriter Terminals) and 713s. The subsystem consists of a CP program and several PP programs as follows:

TELEX	-	TTY Executive Initialization Routine. This routine is loaded at 40000B relative to control point 1 when the operator types TELEX. It initializes tables and pointers and loads TELEX1.
TELEX1	-	TTY Executive Processor. This is the main routine that processes I/O for the TTY's. It cracks and processes commands, and makes requests to dump source input to disk and refill output buffers from disk. It communicates with TRANEX (at another control point) to support transaction terminals.
TELEX2	-	TTY Executive Termination Routine. This routine is executed after an abnormal condition is detected or when the operator terminates TELEX with 1.STOP.
1TA	-	TELEX Auxiliary Function Processor. This routine processes functions for TELEX which require PP action.
1TD	-	Terminal Communications Driver low-speed interactive (600 baud or less). It performs communications between TELEX and terminals (accessed via the 6671 and 6676 multiplexers). It also communicates between TELEX and the KRONOS Stimulator (Checkout/Test)
1TO	-	Terminal Input/Output. Called by TELEX to perform terminal I/O requiring disk accesses.
PFM	-	Permanent File Manager. Called by TELEX to process PF requests.

The relationship between the various system routines and subsystem routines is shown in Figure 13-1.

13.1 TTY OPERATION

The flow of data to or from a TTY and a mass storage device is shown in Figure 13-2. The TTY user enters source statements at a TTY using BASIC or TSRUN, etc. These statements are built character by character and stored in POTS (a POT is an eight word buffer) by 1TD.

Whenever 1TD has filled VIPL pots (level-6 CIPL = 2), he issues a dump pot request. TELEX will initiate the routine DMP (local to TELEX) which will call 1TO. In the interim 1TD may have filled another pot. 1TO will dump the VIPL pots onto one sector on MS. Thus, currently, during this phase 20 or 30 words will be written per sector. This is a very inefficient way

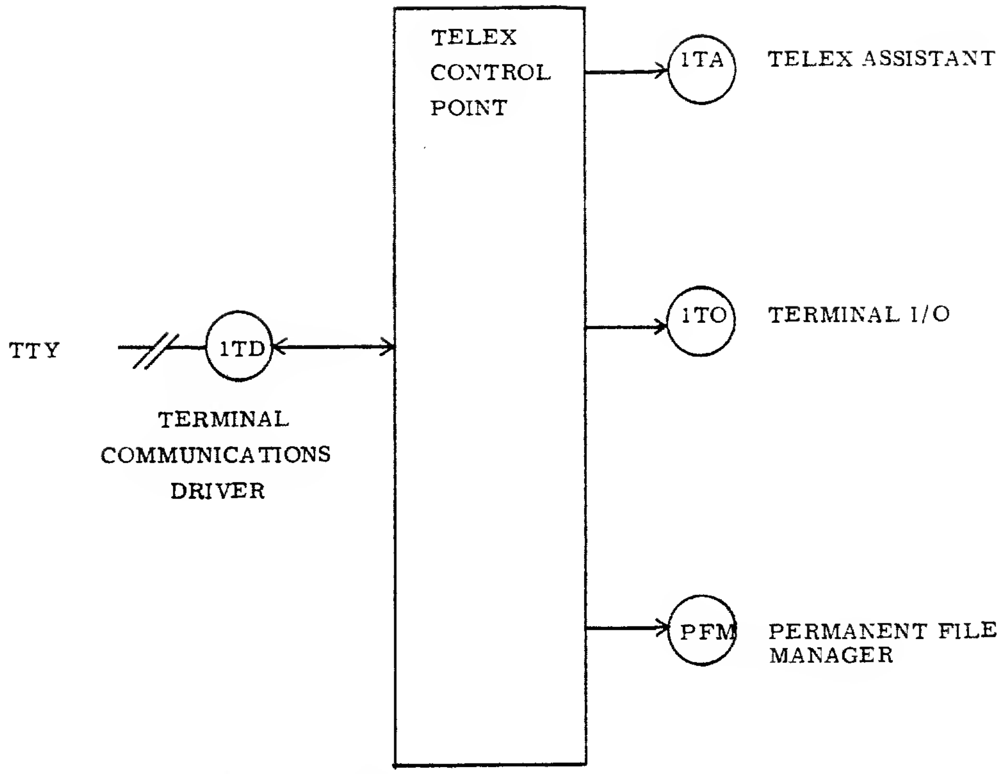


Figure 13-1. TELEX Remote Package

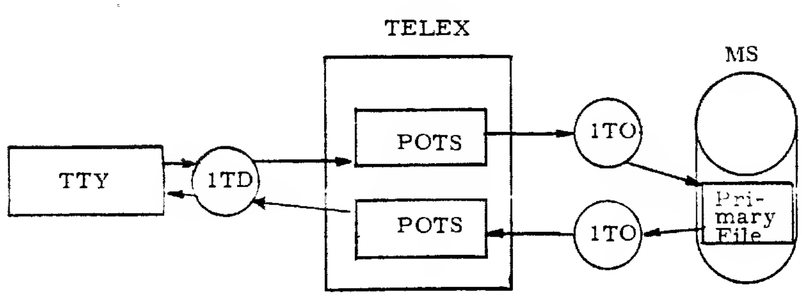


Figure 13-2. TTY Mass Storage Data Flow

to use MS. This will continue until the user enters a command that forces a sort such as RUN or LIST. If the unsorted file is too large, then the message FILE TO LONG TO SORT is issued. In this case, the user must issue the SORT command.

If, however, the file is not too long, then the terminal is placed in sort mode. An MTOT job called MSORT is generated and all users in sort mode will be sorted at once. These users are queued up until a specified time interval has expired, then the MSORT job is run. All the files are given to MSORT in file size order, largest first.

MSORT is an in core SHELL sort. It is started at a CP with the FL necessary to sort the largest file. It will sort the file and rewrite the file in packed format (i.e., 100 words per sector). When MSORT has finished a sort, it will release FL down to the necessary size for the next file and will then sort it. This continues until all the files have been sorted. When MSORT ends, it will be rolled back to TELEX via 1RO. 1RO will set all the terminals whose files were sorted to active mode and TELEX will then process the command that indirectly cause the sort.

This command causes the system to compile a user's job and executes the generated code.

13.1.1 TTY Job Initiation

Refer to Figure 13-3 for this discussion. Assuming that a user's Primary File has been sorted and RUN is typed on the TTY, the following sequence of events occurs.

1. TELEX builds a control card in a POT and calls 1TA. (actual control card is: $LDC,....parameters......) This will be detected later by 1AJ to load the compiler.
2. 1TA builds a ROLLIN queue entry in the system FNT/FST area. The FNT entry will point to the user's rollout file (shown in Figure 13-30).
3. Some time later, the scheduler, 1SJ, will determine that this is the "best job" to initiate, so calls 1RI to rollin the job.
4. 1RI reads the rollout file to build system FNT entries as specified, builds an FNT entry for the Primary File (input to compiler) and initializes a control point.
5. 1RI then calls 1AJ to advance the job which detects the $LDC control cards and loads the compiler with sufficient field length to compile the source statements. After compiling, the program is executed As the job executes it interacts with the TTY by issuing output and receiving input. This interaction is discussed subsequently under "TTY Job Interaction - Output/Input".

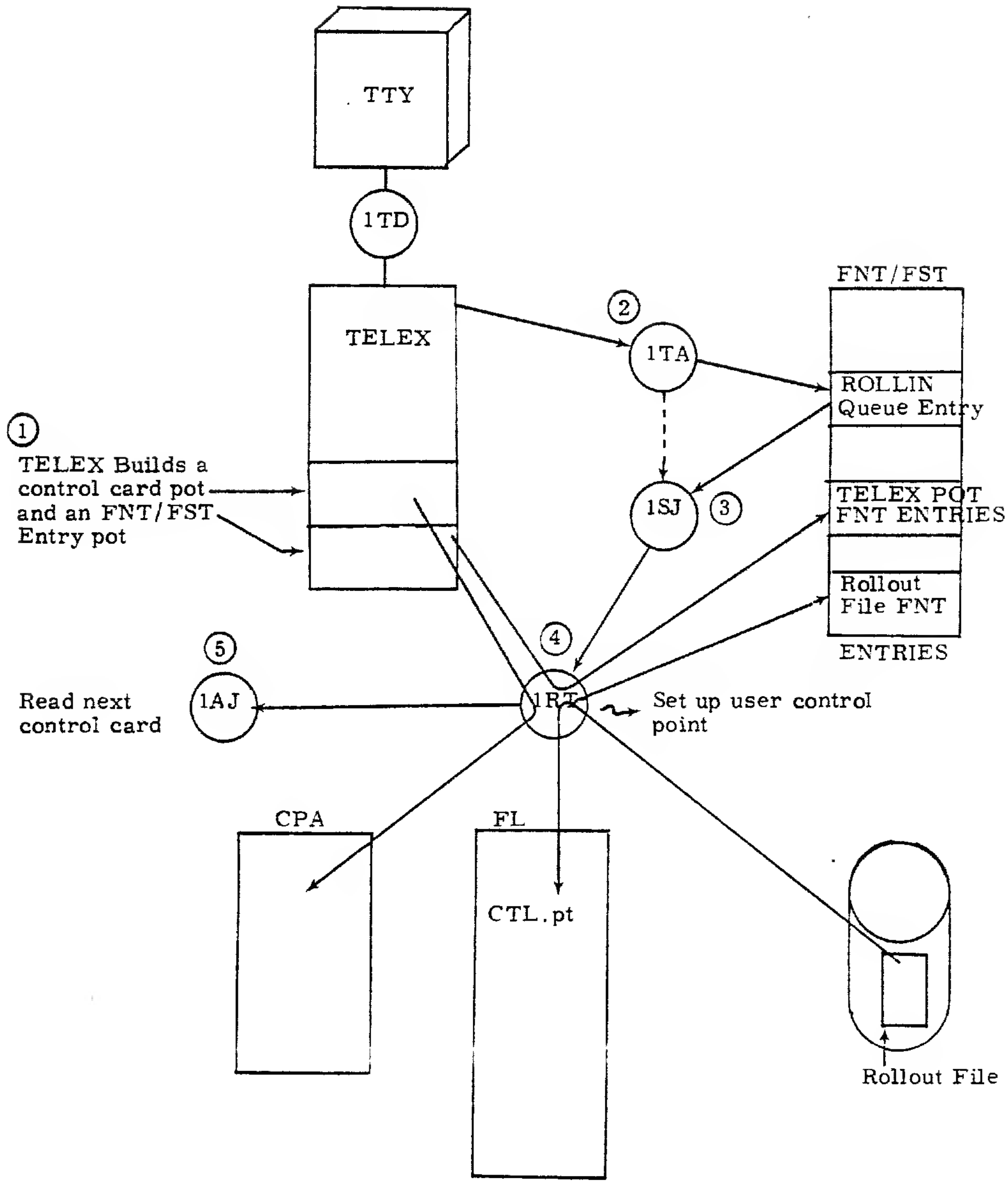


Figure 13-3. TTY Job Initiation

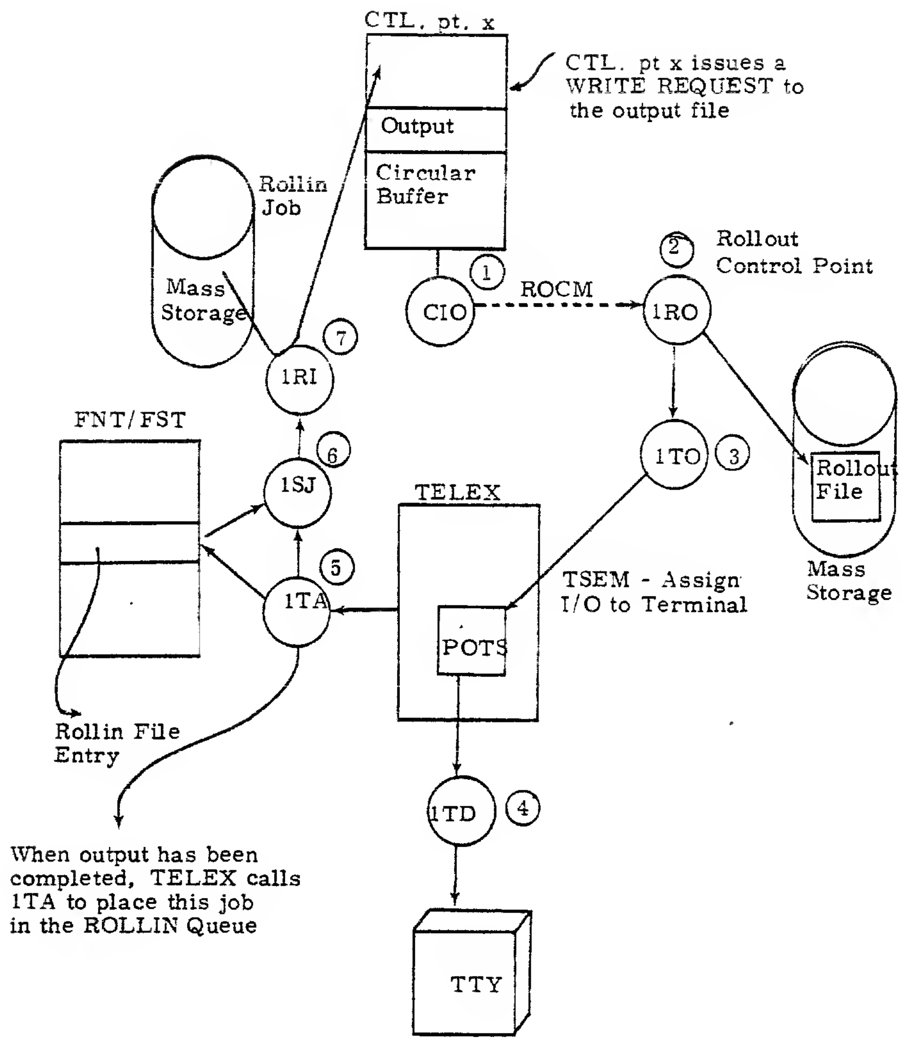


Figure 13-4. TTY Job Interaction (Output)

13.1.2 TTY Job Interaction - Output

Refer to Figure 13-4 for this discussion.

1. CIO is called when the Interactive program issues a write request to the Output file. CIO senses that this is a time-sharing job (TXOT) and issues monitor function ROCM to rollout the control point.
2. Some time later, 1RO initiates the rollout and copies the entire field length (including output data) to the rollout file. In addition, all FNT entries associated with this control point are removed from the system FNT area and stored on the rollout file. Prior to calling 1TO, 1RO reads the first sector of output into 1RO's PP memory where it can be picked up by 1TO without additional disk input/output.
3. 1TO is loaded into the same PP as 1RO. The monitor function TGPM assigns 1TO POTs which store the output data. 1TO then informs TELEX that output is available for the TTY by issuing monitor function TSEM.
4. 1TD is called by TELEX to transmit the output data in the POTs to TTY. 1TD continues to ask TELEX for additional output and TELEX in turn calls 1TO until all output has been transferred.
5. After all output is transferred, TELEX calls 1TA to reinitiate the time-sharing job. 1TA builds the Rollin file entry in the system FNT area as discussed previously.
6. Scheduler 1SJ selects this queue entry as the "best job" as previously discussed and calls 1RI.
7. 1RI rolls the job into a free control point as discussed previously and the time-sharing job continues to execute.

13.1.3 TTY Job Interaction - Input

Refer to Figure 13-5 for this discussion. Assuming that the time-sharing job is to receive data (input) from TTY, the system performs the following functions.

1. The job issues a read request on the Input file which calls CIO. CIO issues monitor function ROCM to rollout the job.
2. Some time later, 1RO is loaded to perform the rollout operation. 1RO then calls 1TO.
3. ITO issues monitor function TSEM to inform TELEX of the requested input.
4. TELEX calls 1TD to issue the input prompt character " ? ".
5. 1TD stores characters in POTs as they are received from the TTY.
6. When the TTY carriage return is sensed, TELEX calls 1TA to reinitiate the time-sharing job. 1TA builds a rollin queue entry.
7. 1SJ selects the queue entry as the "best job" and calls 1RI.
8. 1RI rolls the job into an available control point and transfer the Input data from the POTs to the job's circular buffer. The job is then initialized (given the CPU) and continues to execute.

13.1.4 TELEX Interactive Job Names

Whenever a job is initiated at a CP, 1TA will generate a job name based on the terminal number and UI of the user. The common deck COMPGJN generate job name is used for this task. Whenever a job is rolled back to TELEX by 1RO, the job name must be decoded back to the terminal number. 1RO uses the common deck COMPGTN generate terminal number for this task. In this way, 1RO knows which terminal table in which to indicate the rollout back to TELEX. The terminal number is coded into the fourth through seventh characters of the job name. The UI is coded into the first thru fourth character. The fourth character then does double duty as part UI and part terminal number.

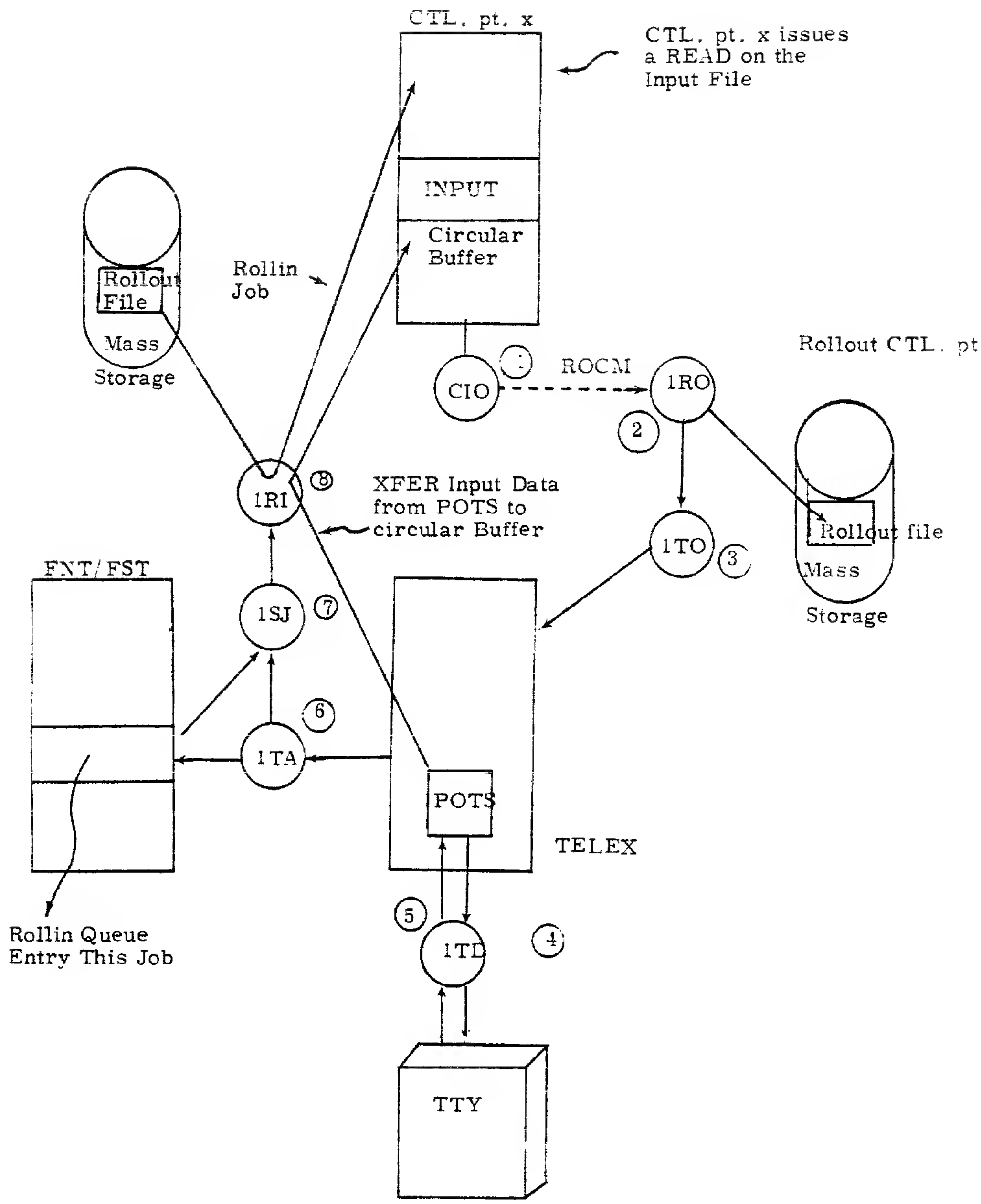


Figure 13-5. TTY Job Interaction (INPUT)

13.1.5 Interactive COMPASS (Program Example)

```
GET,BATFILE
READY.

LIST,F=BATFILE

 71/04/08."14.03.36.
PROGRAM   BATFILE

          IDENT     INTER
          ENTRY     START
OUTPUT    FILEC     OUTBUF.65,(FET=6)
OUTBUF    BSSZ      65
OUT       DIS       10, THIS PROG. CHECKS ON INTERACTION
INPUT     FILEC     INBUF,65,(FET=6)
INBUF     BSSZ      65
IN        BSSZ      20
SETUP     VFD       42/OLOUTPUT,18/OUTPUT
          VFD       42/OLINPUT,18/INPUT
START     SA1       SETUP
          SA2       A1+1
          BX6       X1
          BX7       X2
          SA6       2
          SA7       3
          SX6       0
          SA6       4
          WRITEH    OUTPUT,OUT,10
          READH     INPUT,IN,10
          WRITEH    OUTPUT,IN,10
          WRITER    OUTPUT,R
          ENDRUN
          END       START
READY.

REWIND,BATFILE

READY.

BATCH,50000
/COMPASS(I=BATFILE,B=BATBIN,L=0,0=0)
 ASSEMBLY COMPLETE.
/BATBIN.
  THIS PROG. CHECKS ON INTERACTION
 ? CALGARY IS IN ALBERTA
 CALGARY IS IN ALBERTA
 BATBIN.
 /FORTRAN
 READY.
```

13.2 TELEX INITIALIZATION

Basically, TELEX initializes tables and pointers, then loads and starts TELEX1, the main routine. PP programs called during intialization include:

CIO - Combined Input/Output
CPM - Control Point Manager
LDR - Load Overlay
LFM - Local File Manager
1MA - Issue Dayfile Message
1TA - Auxiliary Function Processor
1TD - Terminal Multiplexer Driver

When the operator types TELEX., DSD calls 1DS which calls 1TD into a PP. 1TD sets the following control cards into the control card buffer:

```
TELEX.
TELEX2.
EXIT.
TELEX2.
```

1TD then calls 1AJ to process the next (first) control card by using the "DIS flag" in the request. 1AJ picks up the first control card, TELEX, has it loaded, and starts the job. After sensing that the driver is ready (at IN13), TELEX allocates tables and establishes the pointers listed in Table 13-1.

Table 13-1. Pointer Addresses

Word	Name	59	47	35	23	11–0
RA+3	VTTP	0	FWA TERMINAL TABLE		LWA+1 TERMINAL TABLE	
RA+4	VPLP	0	FWA POT LINK TABLE		LWA+1 POT LINK TABLE	
RA+5	VCTP	0	FWA COMMUNICATION TABLE		LWA+1	
RA+6	VBMP	0	0		FWA BUFFER MEMORY	
RA+7	VWMP	0	FWA WARN MESSAGE		FWA HEADER MESSAGE	
RA+10	VRAP	0	FWA TELEX REENTRY TABLE		LWA+1	
RA+11	VFNL	Default Family Name				
RA+12	VPTP	0	FWA TRANSACTION		LWA+1 WORD TABLE	
RA+13	UTRN	0	RECEIVE FROM TRANE X		SENT TO BUFFER	
RA+14	DBUG	0	A		B	C

DBUG is the driver debug word

A = driver (1TD) minimum cycle time
B = Moved to C each driver scan
C = { 0 driver scan continues
{ not zero driver scan stops

So, DBUG can be used to debug the driver.

Table 13-1.1 Pointers

Word	Pointer	Description
RA+17	VPPL	NUMBER OF TIMES HAD TO WAIT FOR A PP.
RA+20	VTNL	TOTAL NUMBER OF USERS SINCE INITIALIZATION.
RA+21	VANL	NUMBER OF USERS ACTIVE CURRENTLY.
RA+23	VMNL	MAXIMUM NUMBER OF ACTIVE USERS.
RA+26	VCPL	CONTAINS NEW AVAILABLE POT COUNT DURING THE FL CHANGE.
RA+27	VRLL	NON-ZERO INDICATES THIS IS A RECOVERY LOAD.
RA+31	VABL	ABNORMAL OCCURENCE COUNT.
RA+32	VPLL	BYTES 1,2 = MINIMUM NUMBER OF SPARE (POTS/4), BYTES 3,4 = MAXIMUM NUMBER OF SPARE (POTS/4).
RA+36	VPAL	COUNT OF POTS AVAILABLE.
RA+37	VPUL	COUNT OF POTS IN USE.
RA+44	VDRL	DRIVER PARAMETER AREA. (4 WORDS)
RA+50	VTRP	FWA OF MONITOR QUEUE FOR *TSEM*.
RA+60	VTGP	FWA OF MONITOR QUEUE FOR *TGPM*.
	MUXP	TELEX MULTIPLEXER TABLE

Table 13-1.2 Constants (Lev 4)

Constant	Value	Description
VTTL	10	LENGTH OF EACH TERMINAL TABLE ENTRY.
VDSL	100	LENGTH OF DRIVER CIRCULAR STACK
VSPL	20	MINIMUM NUMBER OF SPARE POTS PER 64 USERS.
VMPL	40	MAXIMUM NUMBER OF SPARE POTS PER 64 USERS.
VOPL	3	NUMBER OF POTS ISSUED ON REQUEST.
VIPL	2	NUMBER OF INPUT POTS ALLOWED BEFORE DUMPING.
VTRL	10	NUMBER OF WORDS IN MONITOR QUEUE *TSEM*.
VTGL	3	NUMBER OF WORDS IN MONITOR QUEUE - *TGPM*.
VCPT	1	*TELEX* CP NUMBER.
VJIR	2	JOB IN SYSTEM
VRIR	4	JOB TO BE ROLLED IN AGAIN
VIPR	10	INPUT REQUESTED
VOPR	20	OUTPUT DATA AVAILABLE
VCPC	10	NUMBER OF WORDS PER POT.
VDPO	2000	DROP POTS.
VASO	2001	ASSIGN OUTPUT.
VMSG	2002	TERMINAL MESSAGE
VSDT	2003	SET DISABLE TERMINAL CONTROL
VCDT	2004	CLEAR DISABLE TERMINAL CONTROL
NULS	0	NULL SYSTEM.
BASS	1	BASIC SYSTEM.
FORS	2	FORTRAN SYSTEM.
EXES	4	EXECUTE SYSTEM.
BATS	5	BATCH SYSTEM.
ACCS	6	ACCESS SYSTEM.
MSYS	7	MAXIMUM NUMBER OF SYSTEMS.
UTIS	10	DEFAULT USER TIME LIMIT/10.
MTIS	777	MAXIMUM TIME LIMIT/10 ALLOWED A USER.
VPST	4	NUMBER OF PSEUDO TERMINAL TABLE ENTRIES
SCPT	1	SCHEDULING PSEUDO TERMINAL NUMBER
SOPT	2	SORT PSEUDO TERMINAL NUMBER
VSBL	110/VCPC	TRANSACTION SEND BUFFER LENGTH IN POTS
VRBL	110/VCPC	TRANSACTION RECEIVE BUFFER LENGTH IN POTS
MPLT	120B	NUMBER OF PLT WORDS PER 64 USERS ON IN PRIVILEDGED
WCQT	1	COMMANDS
WCQT	100	WAIT COMPLETION QUEUE DELAY TIME (MSEC.)
LIAA	4	LOG IN ATTEMPTS ALLOWED
CBASE	0	DEFAULT BASE FOR COMMAND PARAMETER (OCTAL)
LISDL	2	LIST DELAY TIME
COMDL	6	COMPILE DELAY TIME
EXEDL	5	EXECUTE DELAY TIME
CATDL	5	CATLIST DELAY
SORDL	2	SORT DELAY TIME
BATDL	4	BATCH TIME DELAY
RESDL	4	RESEQUENCE DELAY
SWPDL	0	SWAP IN DELAY
NULDI	10	NULL INPUT RESPONSE DELAY TIME
BASDI	4	BASIC INPUT RESPONSE DELAY TIME
FORDI	4	FORTRAN INPUT RESPONSE DELAY TIME

Table 13-1.2 Constants (Lev 4) (Continued)

Constant	Value	Description
EDIDI	2	EDIT INPUT RESPONSE DELAY TIME
EXEDI	4	EXECUTE INPUT RESPONSE DELAY TIME
BATDI	5	BATCH INPUT RESPONSE
ACCDI	10	ACCESS INPUT RESPONSE DELAY TIME
SYSDI	3	SYSTEM PROCESSED COMMANDS
SORFL	4100B	SORT BASE FIELD LENGTH
MSORFL	4	MULTI - TERMINAL SORT BASE FIELD LENGTH
SALTO	3	SALVARE FILE TIME CHECK (MINUTES)

After initializing the tables, TELEX modifies addresses in TELEX1 code which use the increment instruction OPDEFs. Next, each terminal table entry is set to 'COMPLETE' status by setting VROT = 3 in each entry. Next, VWMP, the warn message address is set to the normal header: KRONOS TIME SHARING SYSTEM - VER. 2.1. Next, TELEX calls 1TA to search for time-sharing jobs in the system The jobs searched for are TXOT and MTOT type. The count of such jobs is returned in a pseudo terminal table for TELEX If the count is non-zero, TELEX aborts with the message: TELEX INITIALIZATION ABORT. Next, each driver queue is initialized by setting FIRST, IN, OUT, and LIMIT. Indeed, the driver queues are used like circular buffers. Finally, after starting the drivers and initializing the recovery file (SALVARE), TELEX is complete and control is given to TELEX1 by an EQ jump to TEL.

13.3 TELEX1 - MAIN PROGRAM

TELEX1 is the main program that controls and coordinates the time-sharing subsystem. This program is driven by the following queues:

Request Entering TELEX:

- Driver Request Queue - Requests from 1TD
- Monitor Request Queue - Requests from other PPs
- Monitor Pot Request Queue - Requests from other PPs for pots

Internal Control:

- Wait Completion Queue - Wait for completion of a process
- Time Delay Queue - Wait for time to elapse
- Job Queue - Wait to do all job scheduling at one time
- Sort Queue - Wait to do all sort scheduling at one time

Requests sent by TELEX:

- 1TA Queue - Send all 1TA requests at one time
- 1TO Queue - Send all 1TO requests at one time
- PFM Queue - Send all Permanent File Requests at one time

These queues are scanned by the TELEX1 control loop which is defined in the TELEX flow chart of Figure 13-6.

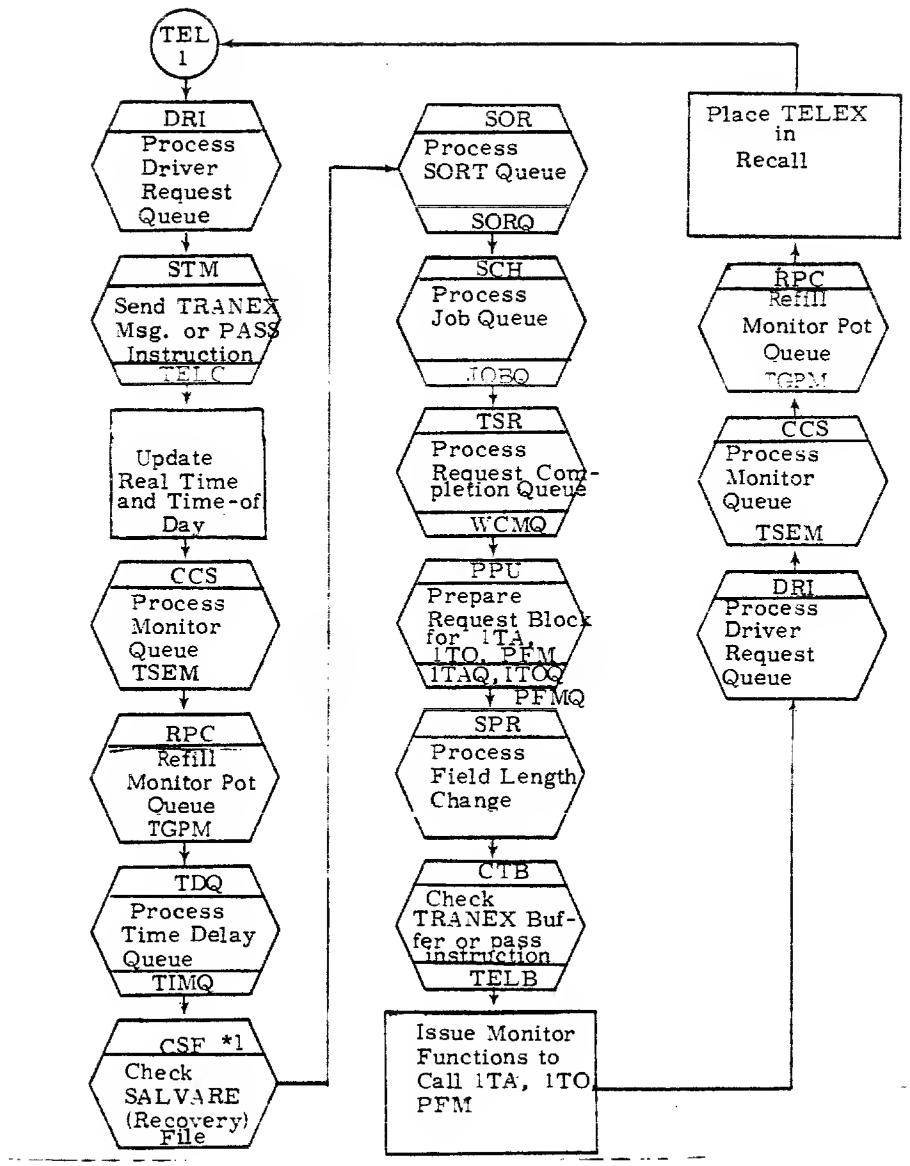


Figure 13-6. TELEX1 Control Loop

*1 The format of the SALVARE file is:

It is two - one word entries per port.

The rollout file contains the Terminal Table entries. For any activity, it is updated. Every 3 minutes the SALVARE file is checked and if the time is over 10 minutes old, the entry is removed, the rollout file dropped, and the terminal logged off if still connected in the Terminal in READ mode. If the terminal is dropped because of a system failure and a new user logs onto the same port and he also is dropped, then the file contains up to two users recoverable per port. Any others are lost. The users must recover within 10 minutes of system recovery or their SALVARE file entry will be eliminated. If the system does not recover until 10 minutes or more have lapsed, the users must log in within 3 minutes to recover. See example of the SALVARE file at the end of this chapter in section 13.9.

The relationship between processing modules of TELEX1 is shown in Figure 13-7.

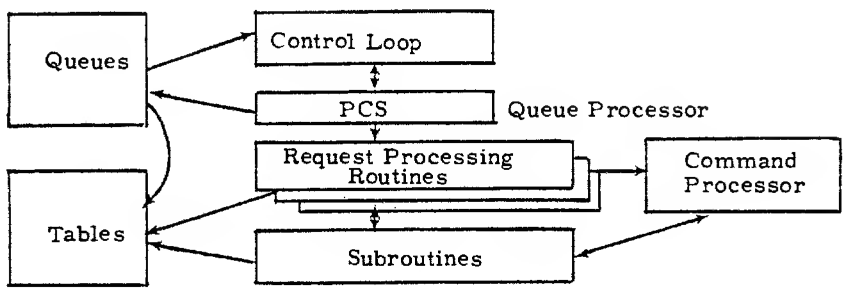


Figure 13-7. TELEX1 Processing Modules

In general, all tables in TELEX are dynamic in length at initialization time. The lengths of the various tables and queues are determined by the maximum number of terminals to be serviced. Thus, it is necessary for all routines at initialization time to determine the values of table pointers, etc. Once TELEX is initialized, the lengths of tables do not change. Thus, pointers such as FIRST and LIMIT could be read and saved by programs that are time critical. These pointers could also be saved as abolute addresses because TELEX will never pause for a storage move. Thus no SYSEDITs should be run while TELEX is running. TELEX1 memory layout is shown in Figure 13-8.

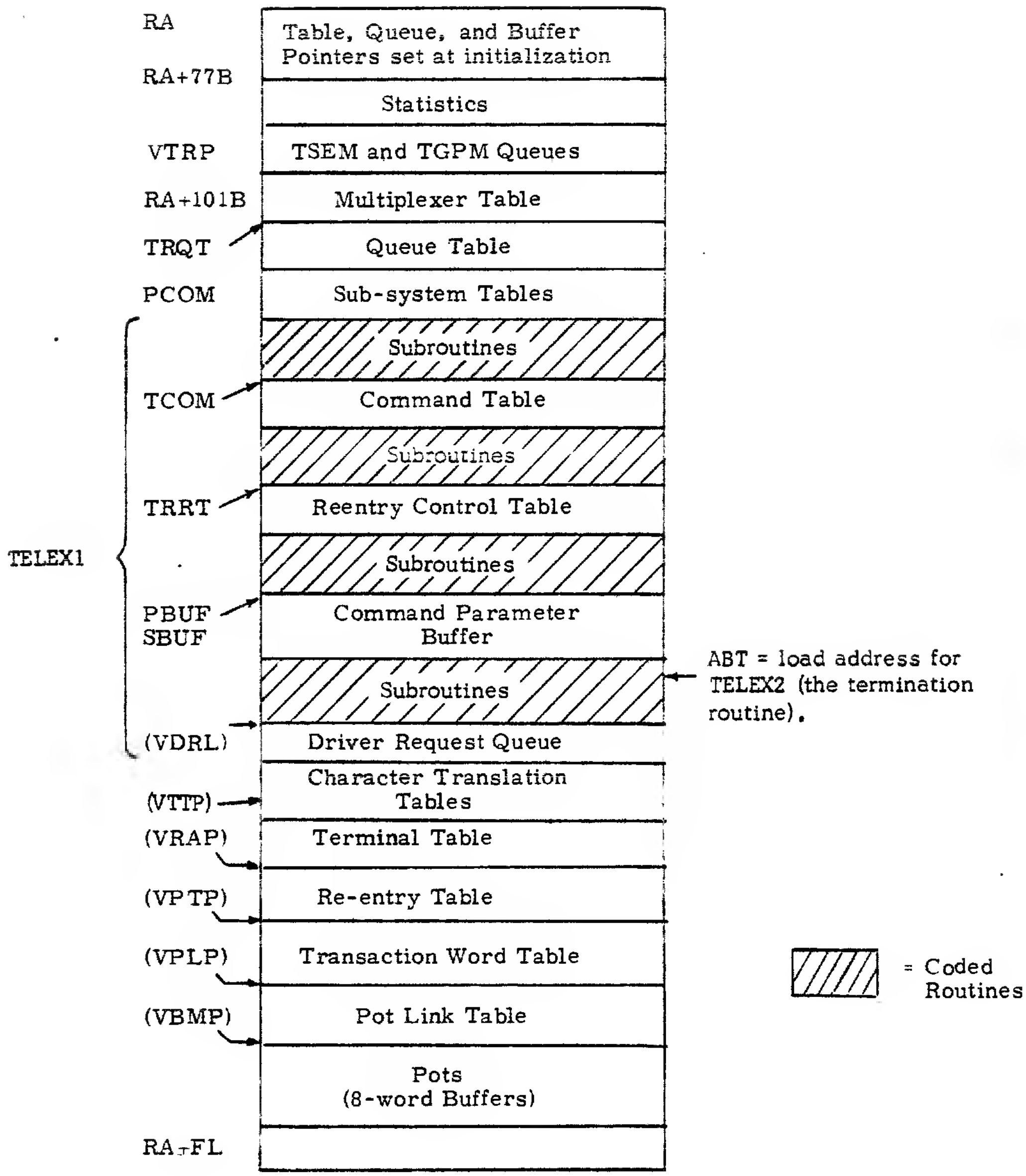


Figure 13-8. TELEX1 Memory Map

13.3.1 Driver Request Queue(s)

Driver (1TD) Requests are passed to TELEX1 via the Driver Request Queue which are circular stacks as shown in Figure 13-9.

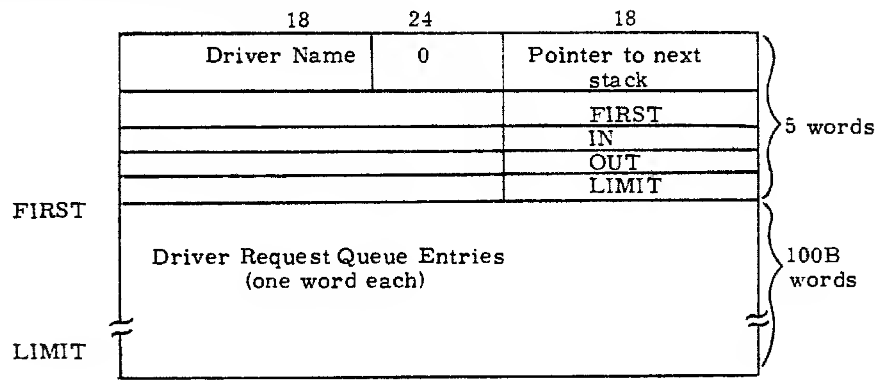


Figure 13-9. Driver Request Queue Stack

Driver Request Queue Entries are placed in a circular stack by 1TD. IN pointer is updated by 1TD when an entry is placed into the queue. TELEX1 updates the OUT pointer as the driver requests are completed. The driver name is stored in word 1 with a pointer to the next stack. A zero pointer indicates the last stack. Each stack is 105B words long (100B words for entries +5 header words). A maximum of eight stacks exist; one for each driver (1TD). The entries are one word each as shown in Figure 13-10.

59 48	47 36	35 24	23 12	11 0
2000+RQ	0	P2	P1	TN

RQ = Request Number
P2 = Parameter 2
P1 = Parameter 1
TN = Terminal Number

Figure 13-10. Driver Request Queue Entry

The request number is always biased by 2000B so that a jump table index can be stored in a B register with use of the unpack instruction. For example, if the above word is in X2, consider the instruction:

```
UX1,B7  X2
```

The result is that B7 contains the request number and X1 contains the parameters and terminal number (i.e., the lower 48 bits). A list of request numbers (request codes) is maintained in common deck COMSTDR and are listed in Table 13-2.

TABLE 13-2. DRIVER REQUEST NUMBERS (Issued to TELEX)

Request Code	Symbol	Description
0	AOD	Increment retry count
1	CSC	Circuit scan complete
2	CLI	Command line input, P1=first pot, P2=word in pot
3	DIN	User dialed in
4	DLO	Data lost, P2=type
5	DRP	Drop pot,
6	DRT	Drop pot chain, P1=first pot
7	HUP	User hung up phone
10	IAM	Issue accounting message, P2=type
11	ITM	Issue terminal message, P2=message number
12	LOF	Log off user
13	LPT	Request additional pot, P1=current pot
14	MAL	Set transaction terminal malfunction P1=status (1=malfunction, 0=O.K.)
15	MIN	Terminate monitor mode (monitor teletypewriter)
16	RES	Request more output, P1=current pot
17	RIN	Release source line, P1=first pot
20	SKY	Interrupt from terminal, P2=interrupt level
21	SPT	Set transaction output pot, P1=pot
22	SSC	Set transaction sequence code, P1=code
23	TTI	Transaction terminal input

13.3.2 Monitor Request Queue(s) (for PPs other than 1TD)

PP requests for TELEX processing are handled via the PP monitor function TSEM. The message buffer is set up by the requesting PP according to the format shown in Figure 13-11.

59 48	47 36	35 24	23 12	11 0
2000B - FN	P1	P2	P3	P4

where,

P1 = P4 are parameters depending on the function

FN = function code. These function codes are defined in packed format in common deck COMSREM. They are listed in Table 13-3

Figure 13-11. TSEM Monitor Request Format

TABLE 13-3. COMSREM FUNCTION CODES

Name	Value	Description
VDPO	2000	Drop pots
VASO	2001	Assign terminal output
VMSG	2002	Assign terminal message
VSDT	2003	Set "disable terminal control" flag
VCDT	2004	Clear "disable terminal control" flag

PP monitor picks up the above request and stores it in a free slot in TELEX's monitor queue for TSEM functions. This queue is located at VTRP in TELEX and is 10B words long. If no slot is free in this queue, monitor (MTR) keeps trying until TELEX honors an existing request and clears a slot.

In general, TELEX drops any unused pots in the chain. If the last pot is not completely filled by the routine issuing output, the routine must put in a terminator byte (0001) in the output data.

NOTE

When issuing a 2001, terminal status must have bit 2^4 set in VROT.

The parameters for the various functions are shown in Figure 13-12.

VDPO - drop pots; TELEX routine - DRT

2000	0000	YYYY	XXXX	NNNN

where,

YYYY = last pot to be dropped
XXXX = first pot to be dropped
NNNN = terminal number

VASO - assign output; TELEX routine - ASO

2001	0000	YYYY	XXXX	NNNN

where,

YYYY = last pot of output
XXXX = first pot of output
NNNN = terminal number

Figure 13-12. TSEM Monitor Function Parameters

VMSG - assign message; TELEX routine - DSD

2002	0000	YYYY	XXXX	NNNN

Where,

YYYY = last pot of message
XXXX = first pot of message
NNNN = terminal number. If below maximum number of pseudo terminals, then this is a warning message sent to all terminals.

VMSG is used by DSD to process the DIAL and WARN operator commands.

13.3.2.1 VSDT and VCDT TSEM Requests

When a TTY user initiates a CP program, the TTY reserves the right to terminate that program with the S or STOP entry. If the CP program wishes to disable/enable this function it can use the DISTC macro described on p. 7-155 of the Reference Manual. This macro generates an RA+1 call to the PP routine TLX. (Notice that if the QP > MXPS+1, this will be interpreted as a CPUMTR function). TLX will issue the appropriate TSEM request function 2003 or 2004, which will set the terminal interrupt address in TIAW as follows. The disable function will ignore this field, and set the disable bit in the terminal table VSTT. The enable function will set this field to the address relative to RA specified in the call and clear the disable bit in the terminal table VSTT. The address is:

1. If INT not specified, then the address is where control is transferred if an S or STOP is sensed on the TTY.
2. If INT is specified, then the address points to a 20B word block where the CP programs exchange package is stored. Control is then transferred to address +20B. The CP routine then can issue an XJR to continue from where it was interrupted. In both cases, if S or STOP is sensed, when control has been transferred, the interrupt address is cleared, so a new DISTC request must be issued.

Figure 13-12. TSEM Monitor Function Parameters (continued)

Pots for output are obtained by issuing the monitor function TGPM. The requests are handled by TELEX in a 3 word queue similar to TSEM requests.

	12	12	36
Call:	12	48	
OR=	TGPM		0
Return:	12	12	36
OR=	0	P	0

P = pot pointer, 0 if no pots available.

If P=0, PPU should reissue the request.

The TELEX TGPM queue size is an assembly constant. Currently at Level-6 it is 3 words long. Whenever a PP needs a POT chain it issues the TGPM MTR request. MTR will search the TELEX TGPM queue for a non zero entry. If MTR finds one, it will be the 1st POT of a POT chain. The chain size is an assembly constant and is currently (at level-6) fixed at 3 POTs. This POT chain is assigned to the calling PP and the queue entry is zeroed. If the queue is empty, MTR will issue an RCLM on TELEX.

During TELEXs main loop it will check this TGPM queue and if it finds any empty entries, it will generate a POT chain and place the 1st POT number in the queue.

The mjaor user of TGPM is 1RO, who requests POTs for flushing a TXOT type jobs OUTPUT file. Another user is DSD, who must get a POT chain for the WARN and DIAL messages.

13.3.3 Terminal Table

The terminal table contains an eight word entry for each possible active user. Each entry contains the current status of each port on each multiplexer. These eight-word entries are structured in such a way so as to minimize interlocks between TELEX1 and the various PP routines which read and write them Each word is shown in Figure 13-13 together with the routines that read and write the word.

Word
0 VUIT - written by TELEX1
read by 1TA

59 18	17 0
USER NUMBER	USER INDEX
42	18

1 VFNT - written by TELEX1, 1RO
read by 1RI, 1TA

59 18	17 12	11 0
Primary File Name	Mode	Batch FL
42	6	12

Modes -
Write lockout - bit 0 set
Execute only - bit 2 set

Figure 13-13. Terminal Table Entry Words

2 VFST - written by TELEX1, 1RO, 1TA
read by 1RI/or the primary file

59 54	53 48	47 36	35 24	23 12	11 0
EQ List	EQ Prim-Est. Ord.	First Track Primary	Current Track	Current Sector	POT FNT Pointer
6	6	12	12	12	12

3 VROT - written by TELEX1, 1RO, 1TO
read by 1RI, 1TA, 1TD, 1TO for the rollout file

12	12	12		12	12	12
Word Count	EST of Rollout File	First Track Rollout	A	Field Length	Sub-Status	Status

A — absolute FL flag; if not set then FL is in units of 100B

Sub-status -

LIST (normal) = 0
LIST (EOR, EOF) = 1
STATUS, F) = 2
With input to 1RI, sub-status is formatted:
LLL L00 00I SSS
where,
L = level number
I = interrupt
S = 1 for EOR status
= 2 for EOF status
= 3 for EOI status

Status -	Bit	Value
TELEX in control	0	0
SYSTEM in control	0	1
Job in system	1	0
Job to be rolled in	2	1
Job awaiting input	3	1
Output available	4	1
LIST or STATUS, F	6	1
Multi-terminal	7	1
Suspended	9	1
Purge files	10	1
error on last operation	11	1

Figure 13-13. Terminal Table Entry Words (continued)

The following three words, VDPT, VCHT, and VDCT are used by 1TD to maintain current information for the terminal. The main loop of 1TD will read these three words into PP memory at direct cells DP, CH, and DC corresponding to VDPT, VCHT, and VDCT. When the main loop jumps to the appropriate routines, they will use these direct cells instead of reading from CM. When control is returned to the main loop, it will write these direct cells back to CM if necessary. VDCT is mainly used for communication with TELEX. This word is interlocked by TELEX thusly. If byte 4 is not clear, then this terminal is being processed by 1TD. When byte 4 goes clear, then 1TD is done and TELEX can use the information to continue activity for this terminal.

4 VDPT - written by 1TD only
read by TELEX1 and 1TD

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
First Pot	Current Pot	Pot Position	Control Flags	Routine Address

Byte 0 - First Pot of input line.

Byte 1 - Current Pot of line being processed.

Byte 2 - Position within Pot as follows:

Bits	Meaning
9-11	First word in first pot of input line
8	Input initiated
7	Next input pot requested
4-6	Current word in current pot (0-7)
0-3	Character number in current word (1-12B)

Byte 3 - Control flags as follows:

Bits	Meaning
4-5	Terminal dependent
3	Binary transmission
2	Transparent input
1	ASCII input
0	Odd parity

Byte 4 - The address of the PP driver subroutine which is currently processing the terminal.

5 VCHT - read and written by 1TD only

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Buffer	Character Count	Scratch	Input Character Count	Output Character Count

Byte 0 - During input, buffer holds the upper (even) character of byte until the next character is received at which time both characters (one byte) are stored into a pot. During output, buffer contains the driver subroutine address.

Byte 1 - Total character count of line being processed.

Byte 2 - Scratch and reentry address for polled terminals (TRANEX) type). It most often contains the current input or output character for non-polled terminals.

Byte 3 - Total number of characters received from terminal.

Byte 4 - Total number of characters transmitted to terminal.

Figure 13-13. Terminal Table Entry Words (continued)

6 VDCT - written by TELEX1 and 1TD
read by TELEX1 and 1TD

Byte 0	Byte 1	Byte 2	Byte 3	Byte4
Flags	Control In-formation	AUTO or MONITOR	Access Level	Next Message

Byte 0 - Flags as follows:

Bit	Value	Meaning
48	0001	Tape Mode
49	0002	Auto Mode
50	0004	Text Mode
51	0010	
52	0020	Transaction Mode
53	0040	Monitor Mode
54	0100	Read Data Mode
55	0200	
56	0400	Input Requested
57	1000	User Logged In
58	2000	Interrupt Complete
59	4000	Driver Request from TELEX1 Byte 4

Byte 1 - Terminal Control Information as follows:

Bit	Values	Meaning
0-2	0-7	First word of output line in POT
3-7	0-37B	User defined carriage return delay
8-9	0-1	Line type 0 = Answerback type 1 = Identification type
10-11		Not used

Byte 2 - In AUTO mode, the line number increment.
In MONITOR mode, the terminal number of the terminal being monitored (i.e., the monitoree).

Byte 3 - Access Control Flags = lower 12 bits of access word defined in VALIDUX file for this user. Refer to the Installation Handbook for procedures to establish the access word. There are ten access bits defined in the system.

- CPWC (bit 0) User may change his password
- CTPC (bit 1) User may use the ACCESS commands
- CLPF (bit 2) User may create direct access permanent files
- CSPF (bit 3) User may create indirect access permanent files
- CSOJ (bit 4) User may have system origin capability from any job origin if the Debug option is turned on by the operator
- CASF (bit 5) User may access system files (common)
- CAND (bit 6) User may request nonallocatable devices (for example, magnetic tape units)
- CCNR (bit 7) Allows use of system without entry of charge or project number

Figure 13-13. Terminal Table Entry Words (continued)

- CSRP (bit 8) User may issue auxiliary device commands
- CSTP (bit 9) User may access special transaction functions

Byte 4 - First POT of an output message assignment or Driver Request Function Code (Byte 0-bit 59 flag). (Refér to paragraph 13.5.3 BGI - STT Subroutines)

7 VSTT - written by TELEX1
read by TELEX1, 1TA, 1TD, 1TO

Byte 0	Byte 1	Byte 2	Byte 3		Byte 4
Flags	First Pot Source	Command Index or Pot Count	RES-ERVED	SYS	Queued Output POT pointer

Byte 0 - Flags as follows:

Bit	Value	Meaning
48	0001	Log-out in progress
49	0002	Log-out abort flag
50	0004	Warning issued
51	0010	Run complete message
52	0020	Sort flag
53	0040	Time limit flag
54	0100	Job complete flag
55	0200	Input lost or job not started
56	0400	Not used
57	1000	Charge number required
58	2000	User limits or alternate PF device
59	4000	Disable terminal control

Byte 1 - First pot of source line input. This byte, along with byte 2 (pot count), is used in subroutine DMP to dump POTs to disk as input is received by calling 1TO.

Byte 2 - POT count or index into command table, TCOM. The index is set by subroutine SCT.

Byte 3 - Non-zero if files lost on RECOVER command or,
SYS = current system in control

0 = Null	3 = not used	6 = Access
1 = Basic	4 = Execute	7 = Transaction
2 = Fortran	5 = Batch	

Byte 4 - POT pointer to a queued output message. That is, if a message is already in VDCT and not yet processed, the next message is queued by using byte 4 of VSTT. If another message must be assigned, it will be lost. See subroutine ASM. Normally, this byte is zero.

Figure 13-13. Terminal Table Entry Words (continued)

Table 13-4 is a summary of the terminal table entry.

TABLE 13-4. TERMINAL TABLE ENTRY SUMMARY

Name	Word	Written by	Read by
VUIT	0	*TELEX, 1TA	
VFNT	1	*TELEX, 1RO	1RI, 1TA
VFST	2	*TELEX, 1RO, 1TA	1RI
VROT	3	*TELEX, 1RO, 1TA	1RI, 1TA, 1TD
VDPT	4	1TD	
VCHT	5	1TD	
VDCT	6	*TELEX, 1TD	
VSTT	7	*TELEX	1TA, 1TD

*The name TELEX refers to any of the three overlays comprising TELEX. Any routine which writes a word also is assumed to read that word.

13.3.4 Transaction Word Table

The transaction word table provides TELEX/TRANEX communication and is pointed to by VPTP and contains a one-word entry for each transaction terminal. Figure 13-14. shows the entry format.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
Retry Count	Status Flags	Output Pot Chain	Message Sequence	Terminal Address

Status Flags:

Symbol	Bit	Meaning
UTOB	47	Terminal off
UTMB	46	Terminal malfunction
UAMB	45	Terminal waiting for message
UWOB	44	Terminal waiting for output

Figure 13-14. Transaction Terminal Word.

These words are written by TELEX Transaction routines and read by the driver, 1TD.

The changes to the terminal table for NIXDORF transaction lines are:

VCHT	Byte 1	bit 5-11 time out count
	Byte 2	reentry address (i.e., index into protocol sequence).
	Byte 3	block check character
	Byte 4	terminal number
VDPT	Byte 3	bit 0 on - retry in progress bit 3 on - output sent bit 4 on - input received bit 5 on - sequence error

13.3.5 POT Link Table

The POT Link Table (PLT) controls the use of POTs (8 word buffers). Its layout is shown in Figure 13-15.

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
VPLP+0	7777 0	0002 1	0003 2	0004 3	0017
+1	0005 4	0000 5	0007 6	0000 7	0017
+2	0000 10	0012 11	0013 12	0014 13	0007

Byte 4 - RES = Reservation flags

000 000 001 111

pot in byte 3 reserved; 0 = free POT
pot in byte 2 reserved; 0 = free POT
pot in byte 1 reserved; 0 = free POT
pot in byte 0 reserved; 0 = free POT

Figure 13-15. POT Link Table

Each byte (0-3) represents a POT, an 8-word CM buffer starting at VBMP. Bytes 0-3 contain a link to the next POT in the chain. The last POT in the chain is indicated by a zero byte. POT zero is always reserved and LINKs to 7777. Each PLT byte has the following format:

11 — 2	1 0
word link	byte link

Example: In Figure 13-16 sample table, pots 1-5 are reserved and comprise one chain. POTs 6 and 7 comprise another chain. POT 10 is free. POT 11 is the start of another chain.

13.3.6 Internal Queues (TRQT)

All internal queues are built at assembly time in a "table of queues." This table consists of all the queues that may have requests in the re-entry table. The following is a list of valid queue names in the "table of queues."

WCMQ - Wait Completion Queue
TIMQ - Time Delay Queue
JOBQ - Job Queue
SORQ - Sort Queue

ITAQ - 1TA queue
ITOQ - 1TO queue
PFMQ - PFM queue

} PP request queues

The PP request queues are one-word entries in the "table of queues," while the other 4 are two-word entries. The format of the entries is shown in Figure 13-16.

	12	6	6	12	12	12
One Word	PP	P	00	FC	TN	PP

where:

PPP = 1TA, 1TO, or PFM
FC = Function code
TN = Terminal number
PP = Pot pointer

word 1	2CCC	0000	00NN	NN : 00	YYYY
word 2	0000	0000	TTTT	TTTT	TTTT

where:

CCC = number of entries (packed format)
NNNN = first terminal entry (index into Re-entry Table)
YYYY = last terminal entry (index into Re-entry Table)
T - T = resource control count

Figure 13-16. Table of Queues Two-Word Entry Format and One Word

NOTE

Each queue has an associated string of entries in the Re-entry Table. See Figure 13-19.

13.3.7 Re-Entry Table (VRAP)

The TELEX subroutines use the reentry table to have control returned or functions performed for them when a set of conditions are met. The table consists of one word for each terminal with one of the formats shown in Figure 13-17.

1.

0000	0000	0000	0000	0000

No reentry conditions

2.

2YYY	XXXX	XXXX	PPPP	NNNN

YYY = Index to TRRT (table of reentry processors)
XXXX XXXX = anything
PPPP = POT pointer for further params
NNNN = LINK to next entry in the queue of this type (see TSR)

3.

0000	0000	0000	00	NN NNNN (bits 17–0)

NN NNNN = pot address of stacked entries

Figure 13-17. One-Word Re-entry Table Formats.

Each entry in the Re-entry Table contains an index to the Table of Re-entry Routine Parameters (TRRT).

13.3.8 Table of Re-entry Routine Parameters (TRRT)

This table is built at assembly time. It consists of entries that direct further processing based on entries from the re-entry table and on completion of certain sections Entries are added to the table by use of the COMMAND macro. Entries are one word, according to the format shown in Figure 13-18.

59 48	47 36	35 18	17 0
XXYY	ZZZZ	EEEE EE	NN NNNN

where

XX = index to TRQT (Queue Table). If XX=0, no resources are required except for a peripheral processor, possibly.
YY = function code for called program.
ZZZZ = function processing address relative to TSRPROC.
EEEEEE= error return address.
NNNNNN= normal return address.

Figure 13-18. TRRT Format

The COMMND macro parameters are:

COMMND MACRO PROC, SYSR, NPRO, ERRA, FUNC

ZZZZ=PROC = entry point of routine to process this command.

XX=SYSR = the queue that the request is to be placed in. (WCMQ, TIMQ, JOBQ, SORQ, ITAQ, ITOQ, or PFMQ).

NNNN=NPRO = normal return address.

EEEE=ERRA = error return address.

YYY=FUNC = function code to be passed to the called program

Example of COMMND macro:

This example shows the use of the COMMND macro and how easily a call is made to generate a queue entry.

```
         COMMND INP6,WCMQ,INP6,INP6
INP6$    EQU     *       (This is generated by the COMMND macro).
```

INP6$ is the symbol for this word in the table of reentry routines.

Example of COMMND INP6,WCMQ,REENTI,ERR:

```
               Log in code
LIN            code
                 |
               CALL SPRR                          set up ITA call, now - can't wait for
                                                  response so queue up for this terminal
                                                  and return later
               SX5              INP6$             command address
               EQ               PCSU
                 |
REENTI           v              good return

ERR                             error return
```

Now to make the WCMQ queue entry:

```
      •           •
      •           •
      •           •
     SX5         INP6$    SPECIFY COMMAND TABLE ENTRY
     EQ          PCS4     MAKE QUEUE ENTRY
      •           •
      •           •
      •           •
 INP6 BSS         0       NORMAL AND ERROR RETURN ADDRESS
      •           •
      •           •
      •           •
```

In general, queue entries are made in this manner throughout TELEX

Figure 13-19 shows the relationship between the Table of Queues", the Re-entry Table, and the Table of Re-entry Routine Parameters. Notice that there is one queue entry per terminal.

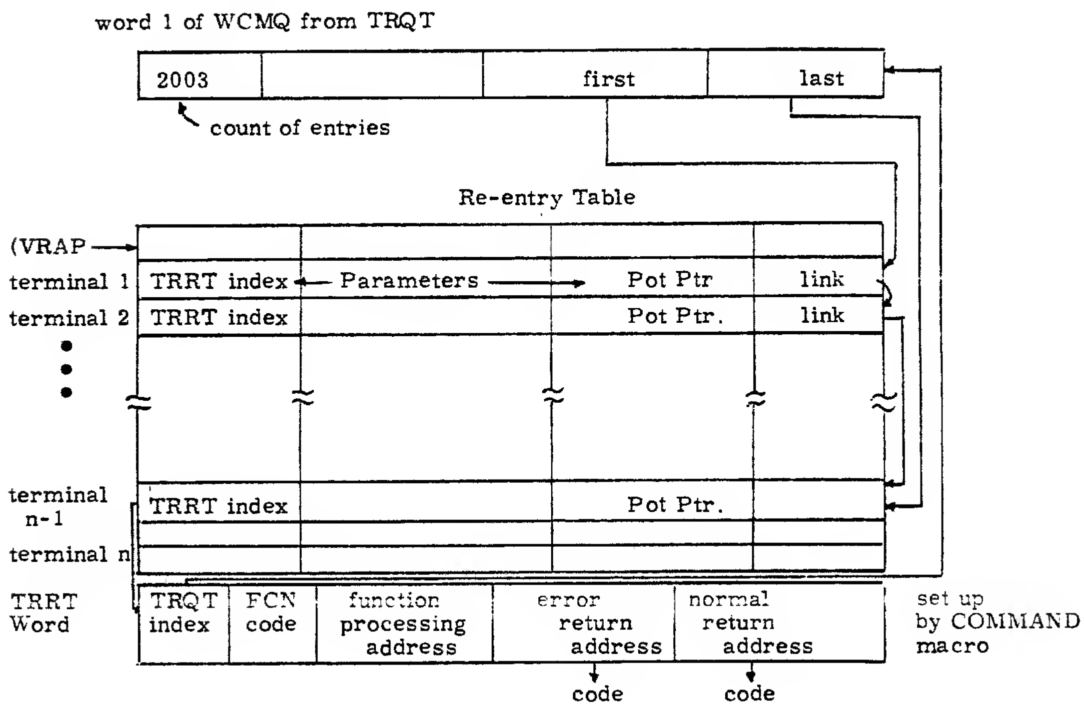


Figure 13-19. Table Relationships

13.3.9 Queue Processing

Processing of queue entries is done by the PCS subroutine. As entries are completed, PCS extracts the normal or error return address and jumps to it. Making queue entries is done by a jump to PCS4 or PCS6. Before returning to a routine, PCS calls SSP which sets up the following registers:

A0 = FWA of user's terminal table entry
B2 = terminal number
B3 = POT pointer (extracted from byte 3 of entry in Reentry Table)
B4 = FWA of pot pointed to by B3
X7 = bits 24-47 of Reentry Table entry

These A and B registers are generally not changed within the various subroutines of TELEX

13.3.10 TELEX Routines

The following is an outline of the subroutines comprising TELEX:

- MUXP - multiplexer table (RA + 101B)
- TRQT - table of queues:

WCMQ	ITAQ
TIMQ	ITOQ
JOBQ	PFMQ
SORQ	

- TEL - control loop. Calls the following:

DRI	TDQ	PPU
STM	CSF	TSR
CCS	SOR	SPR
RPC	SCH	CTB

- CCS - Process requests to handle output to TTY by calling the following subroutines:

DRT	SDT
ASO	CDT
DSD	

- CSF - Check SALVARE file user time out
- DRI - Process driver (1TD) requests by calling the following subroutines:

AOD	DLO	IAM	MAL	SKY
CSC	DRP	ITM	MTN	SPT
CLI	DRT	LOF	RES	
DIN	HUP	LPT	RIN	

- PCM - Process terminal commands (called from CLI, AUT)
  Calls following subroutines:

ACC	DIA	LIS	REP	SUB
ASC	EDI	MTR	PER	TAP
ATT	FDP	NOR	ROT	TER
AUT	GET	NOD	RUN	TXT
BAT	HEL	NOS	SAV	UNS
BIN	HDP	PAC	SOF	UNU
BYE	LAN	PAR	STA	XEQ
CLR	LEN	PFC	STO	

- Reentrant Command Processing Routines:

BJB	IEX	IUA	IAF	PUR
BJS	INJ	PBS	PFF	RDY
EJB	IPF	PSS	PFM	
IDT	IPL	DAF	PFP	

- PCS - Process queue entries
- PPU - Process PPU requests
- RPC - Refill POT chains
- SCH - Build job queue entry for scheduling a job
- SOR - Set up for scheduling SORT job
- SPR - Call 1TA to adjust field length
- TDQ - Process time - delay queue
- TSR - Process WCMQ. Reenter the following:

DCR	ITA	MJE	SRE
HNG	ITO	MTO	SSO
ICH	JOB	REC	
INP	LIN	SEN	

- General Subroutines including:

ABT	CPF	GPL	MQE	SFL
BRQ	DAP	GQE	MVA	SLF
CCM	DMP	GRT	O6S	SRC
CFL	DPT	GTA	PCB	SRR
CJT	ENP	GZP	RPL	SSP
CLE	GEM	ISH	RPT	TPF
COI	GFN	LTT	SAF	UPF
COP	GFS	MDA	SCT	UQS

- Transaction routines including:
  a. TRANEX driver routines
  b. TRANEX interface routine
  c. general subroutines

13.4 TELEX2 - TERMINATION

TELEX2 performs termination procedures for the TELEX subsystems. It is called whenever an abnormal condition is detected or when the operator types 1.STOP to drop the subsystem.

When an abnormal condition is detected within TELEX1 processing, a jump to the abort subroutine (ABT) is executed. If sense switch 3 is OFF, ABT continues or control is returned to the calling routine If switch 3 is ON, ABT issues the message:

TELEX ABNORMAL - XXX

where: XXX is the name of the subroutine calling ABT.

After issuing this message, the ABORT macro is used to abort the control point. 1AJ senses the EXIT control card, the next control card (TELEX2) is found, and 1AJ has the termination routine loaded. Loading of TELEX2 starts at location ABT. This overlays the least important code of TELEX1 and leaves the tables and queues untouched. Basically, TELEX2 logs out all active users so that there will not be any time-sharing jobs left in the system. After issuing system statistics, 1TD is called to restart the time-sharing subsystem depending on sense switch settings.

13.5 MULTIPLEXER DRIVER

1TD performs communication between TELEX and terminals (accessed via the 6671 and 6676 multiplexers) and the KRONOS Stimulator. It has the capability to communicate with most ASCII compatible terminals and correspondence code compatible terminals such as the IBM 2741 and NOVAR 541, 713, NIXDORF terminals, if the multiplexer has the required options installed.

1TD processes up to 512 (10-character/second) terminals. The number of terminals for which performance can be guaranteed will decrease as the terminal speed is increased In any event, the total driver capability is 5120 characters/second The maximum terminal speed which may be accommodated is 60 characters/second.

Terminal communication is processed in a half-duplex mode. A line is generally the unit of transmission in each direction. Interruption of continuous output is provided along with an input line and character deletion facility.

Communication between 1TD and TELEX is accomplished by means of a circular request queue provided by TELEX. 1TD inserts a request in the queue and TELEX removes the request as it is processed.

Terminal control operations for ASCII terminals include:

1. To complete an input line, type the RETURN key. A line feed is not needed, since the driver issues one to the terminal.

2. To delete or ignore an input line, type the ESC key

3. To delete a previously entered character, type the UNDERLINE (BACK ARROW on some Model 33 teletypewriters.

4. To terminate output, type the BREAK key, or the S key.

5. To interrupt output, type the I key. Output may be resumed by typing P followed by RETURN.

Terminal control operations for correspondence code terminals include:

1. To complete an input line, type the RETURN key.
2. To delete or ignore an input line, type ATTN.
3. To delete the previously entered character, type BACK SPACE.
4. To terminate output, type the ATTN key.

1TD consists of two routines: 1TD and 2TD. the 1TD routine is the initialization (and termination) routine that loads the 2TD overlay. The 2TD overlay is normally loaded and executing in the PP while the TELEX subsystem is servicing terminals. Four other overlays are assembled with 1TD. These are the translation tables for the various terminals listed in Table 13-5.

TABLE 13-5. TRANSLATION TABLES OVERLAYS

Overlay	Terminal Type
9JA	ASCII terminal
9JB	Correspondence/text
9JC	Correspondence
9JD	Memorex 1240/APL

Figure 13-20 shows the multiplexer servicing concept as being similar to the hardware slot and barrel concept for peripheral processors. Notice that up to eight multiplexers are serviced by the driver and that each port is allotted a time slice in which the driver performs I/O and required overhead.

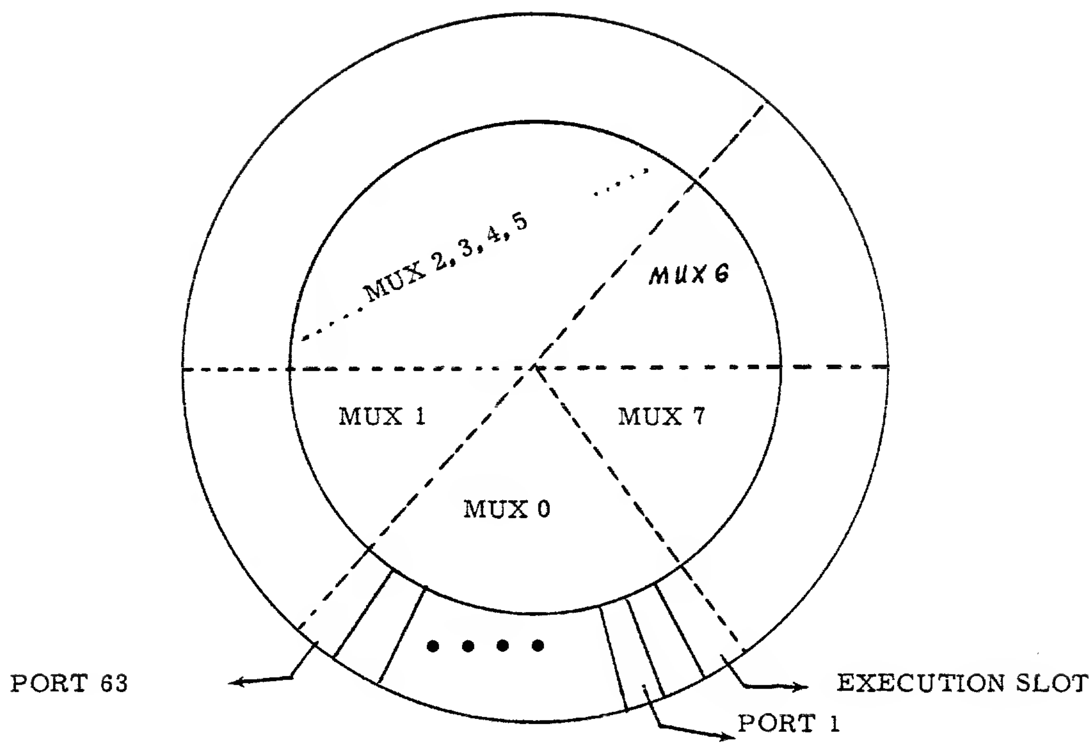


Figure 13-20. MUX Servicing Concept

13.5.1 Driver Initialization (1TD)

The multiplexer driver is initialized by the overlay 1TD. This overlay consists of three USE blocks:

1. MAIN - initialize TELEX control point
2. PRESET - load 2TD
3. RESIDENT - code resident during execution

The lengths of these blocks are determined by the difference between their last word address and their first word address as shown in Table 13-6.

TABLE 13-6. USE BLOCKS LENGTHS

Last	First	Description
MANE	MANF	length of MAIN
RESE	RESF	length of RESIDENT
PRSE	PRSF	length of PRESET

These three lengths are added and the sum is subtracted from 4096 to establish the origination (ORG) address. The multiplexer input buffer (IBUF) is defined in PRESET and must follow the PP resident translation tables. A check for this overflow condition is made at the end of the 2TD overlay. At this time, there are 52B PP words between IBUF and the end of the translation tables. (August 1973)

Overlay 1TD is loaded when the operator types TELEX to 1) (start the time-sharing executive) and 2) (during termination to perform certain post processing operations). That is, 1TD is called by 2TD from the DRP subroutine. Since 1TD is loaded above the translation tables, much of 2TD is overlayed when it calls 1TD. Routines overlayed include some write mode processing (WTM), all polled line processing routines, and all of the utility subroutines. In addition, the translation tables and the multiplexer input buffer are overlayed as well. Figure 13-21 shows the relative load addresses of the three USE blocks comprising 1TD, as well as the 2TD overlay while executing.

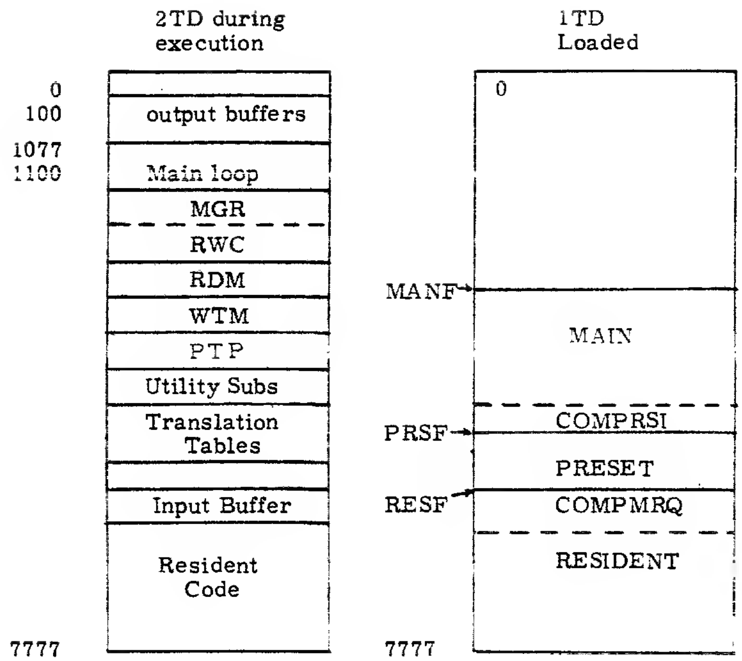


Figure 13-21. 1TD/2TD Memory Maps

Figure 13-22 is a flowchart showing an overview of the initialization processes in blocks MAIN and PRESET. RESIDENT code is used by 2TD during termination processing.

Data in the multiplexer input and output buffers within 2TD consists of an 8-bit character per port along with control bits as shown in Figure 13-23.

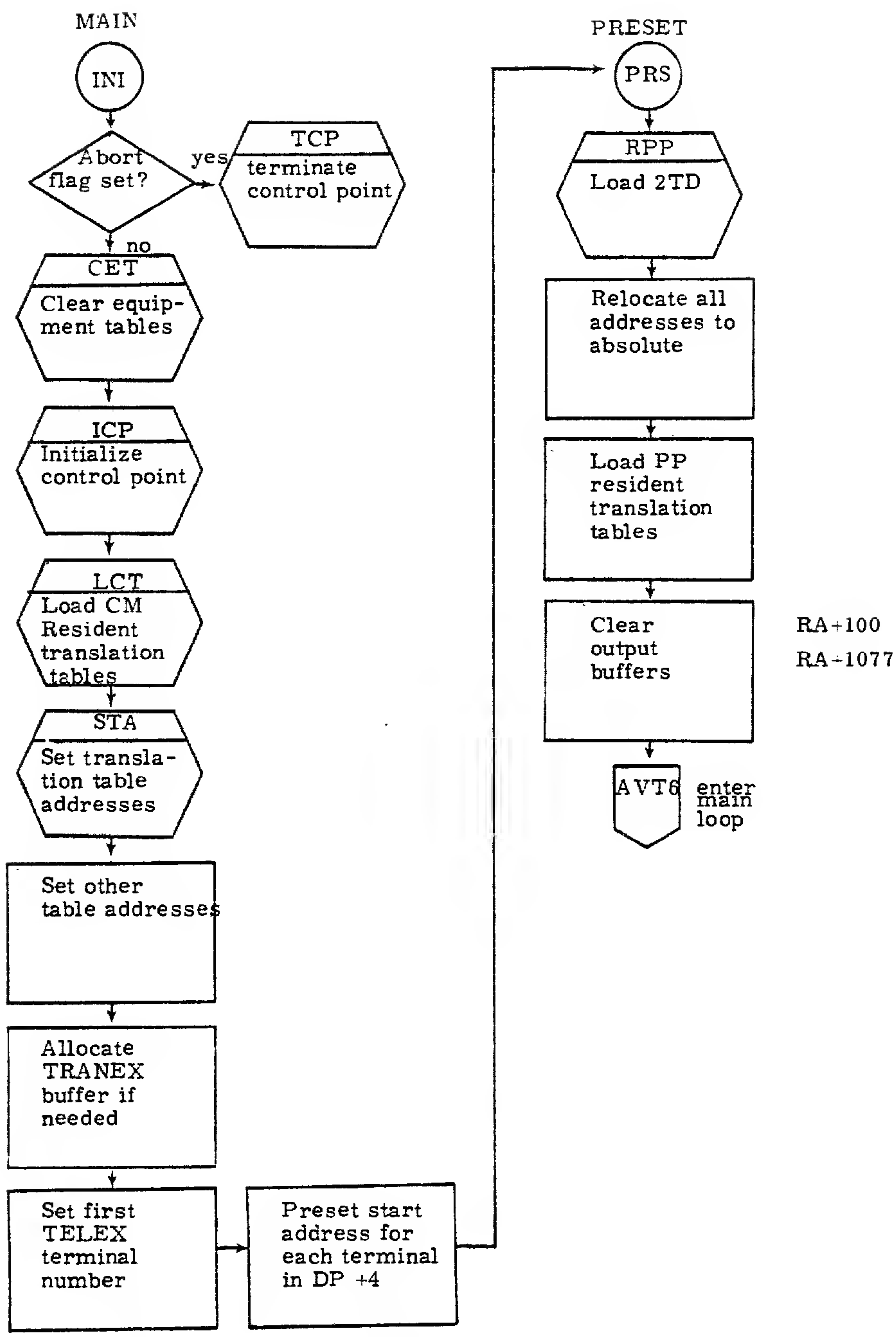
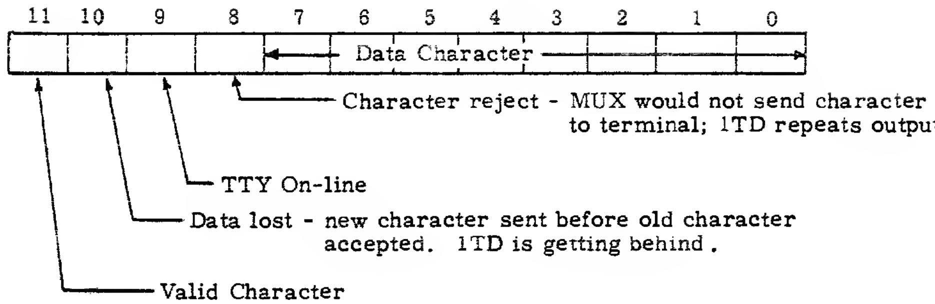


Figure 13-22. MAIN and PRESET Overview

Input from 6676 MUX



Output to 6676 MUX

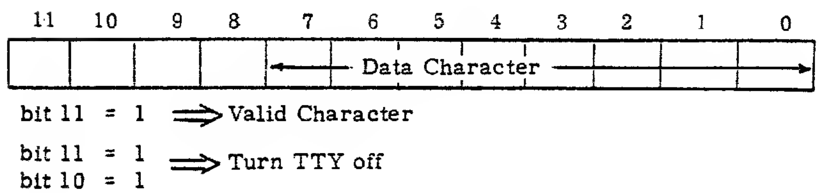


Figure 13-23. Input/Output Buffers.

Note

Further information is available in the following manuals:

Title	Publication Number
Control Data 6671/6671-2 Data Set Controller Ref. Manual	60334600
Control Data 6676-A TTY Multiplexer Ref. Manual	38706000
Control Data 6676-B/C TTY Multiplexer Ref. Manual	38707800

Figure 13-24 describes the logical breakdown of the 2TD driver while executing.

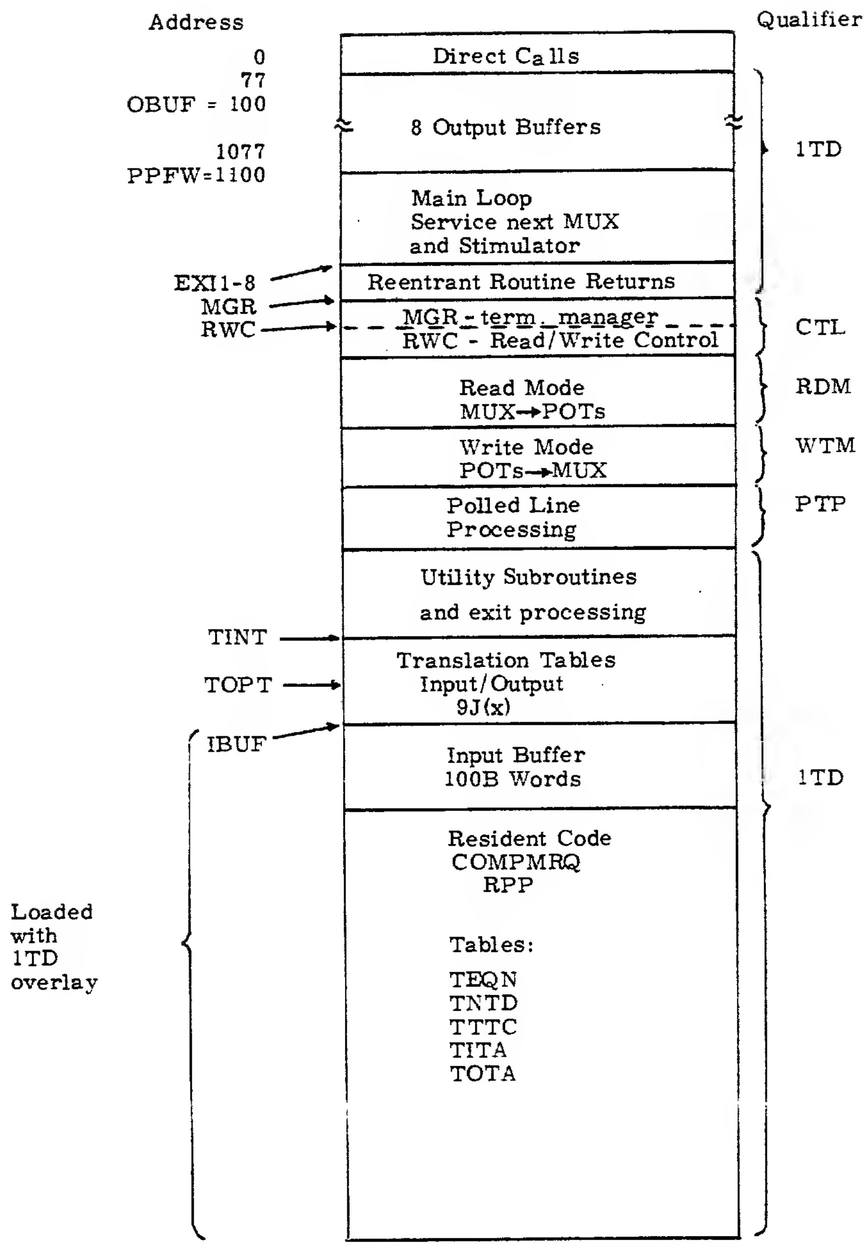


Figure 13-24. 2TD Memory Map

13.5.2 Re-Entrant Routine Returns

The re-entrant routine returns are eight "entry points" which are jumped to by any subroutine which cannot complete its function in a single time slice. The RETURN macro is the most common method used throughout the listing for the purpose of setting a return address Control is returned at the next instruction or to another specified routine address. For instance,

RETURN EXI7

enables control to be returned at the next instruction; while

RETURN EXI3, LIN

causes control to be set to the LIN subroutine for the next time slice for this port. In any case, the EXI(x) specifies a reentrant return address. If x is odd, the reentry address is in the A-register and stored in DP+4 (i.e , VDPT, byte 4). If x is even, no return address is given, and control is returned to the previous return address in DP+4. The reentrant return addresses and terminal table words updated are whown in Table 13-7.

TABLE 13-7. ADDRESSES AND WORDS

Reentrant Return Address	Terminal Table Word(s) Written
EXI1, EXI2	VDPT
EXI3, EXI4	VDPT, VCHT
EXI5, EXI6	VDPT, VCHT, VDCT
EXI7, EXI8	VDPT, clear byte in output buffer

Direct Cell assignments are explained in the listing. However, it is worth noting that during execution VDPT, VCHT, and VDCT are available in direct cells. VDCT is read and updated only when necessary to minimize CM reads and writes.

The main loop controls the advancement to the next multiplexer, performs MUX I/O, checks for STIMULATOR processing, and enters the manager (MGR subroutine

13.5.3 Process Subroutines

The MGR subroutine processes individual ports and satisfies requests from TELEX A flowchart of MGR is shown in Figure 13-25. The symbol qualifier CTL contains the following routines:

MGR - terminal manager
CIS - check interrupt status
INT - process interrupt
CTO - check time out
DIN - dial-in processing
HUP - hang up phone
OFL - process user off line
RWC - Read/Write control
DTT - determine terminal type
LIN - process login
RAB - read answerback drum

1*

ITD - function codes for the processor TFR 2*

TFR - process TELEX functions with the following subroutines:

1 BFI - begin input
2 CFD - clear "full duplex" flag in VDPT ITD function values for BYTE 4 of VDCT.
3 HUP - hang up phone
4 IIP - issue input prompt (i.e., "?")
5 LGI - process login
6 SAS - set ASCII mode flag in VDPT
7 SNM - set normal mode
10 SOP - set odd parity
11 SFD - set "full duplex" flag
12 STT - set terminal type

READ MODE

The symbol qualifier RDM contains the following read mode subroutines:

BRD - binary read
CRD - correspondence read APL type, NOVAR
ARD - ASCII read

These three routines call RTC which translates the input character and stores it in a POT. If the input character is a "special" character, one of the following subroutines is called:

ESC - process escape codes
CRT - process carriage return
DLN - line delete
DPC - delete previous character
NLI - null input
CSF - case shift
NWL - new line
EOT - end of transmission
BRK - break

CRT, BRK, and NWL call EIL for end-of-line processing which calls:

CLI - command line input
SLI - source line input

CLI calls:

ACL - ASCII end of command line
or CCL - Correspondence end of command line

SLI calls:

ASL - ASCII end of source line
or CSL - Correspondence end of source line

1* VDPT word. DP+4 gets one of these address.

2* TELEX requests 1TD to perform certain functions by setting bit 11 of Byte 0 of VDCT and the function code in Byte 4.

General subroutines used by RDM are:

- ITM - issue terminal message
- NIP - no input POT available
- DLO - process lost data
- TIC - translate input character
- WIC - write input character

Normal read mode processing starts with the RDM subroutine which sets the return address in DP+4 to BRD, CRD, or ARD. As characters are received from the multiplexer, they are processed by RTC which calls TIC to translate them, then calls WIC to write them in POTs. The normal exit is to EXI4. Figure 13-26 shows the general relationship of the read mode processing subroutines.

The symbol qualifier WTM contains the subroutines used for write processing. These subroutines are structured similar to RDM subroutines and include:

- BWT - binary write
- CWT - correspondence write
- AWT - ASCII write

These three subroutines call WTC to write the terminal character by using subroutines

- ROC - read output character from pot
- TOC - translate character

A "special" character is processed by one of the following routines:

- NLO - null output
- ANL - ASCII terminal new line
- ACR - ASCII terminal carriage return
- CNL - correspondence end of line
- CCR - correspondence carriage return
- CLF - correspondence line feed
- CBS - correspondence backspace

Other write mode general subroutines include:

- CMM - process monitor mode
- SOC - set output control
- SRC - send repeated character

SOC restarts a job to get more output and processes output control bytes by jumping to one of the subroutines listed in Table 13-8.

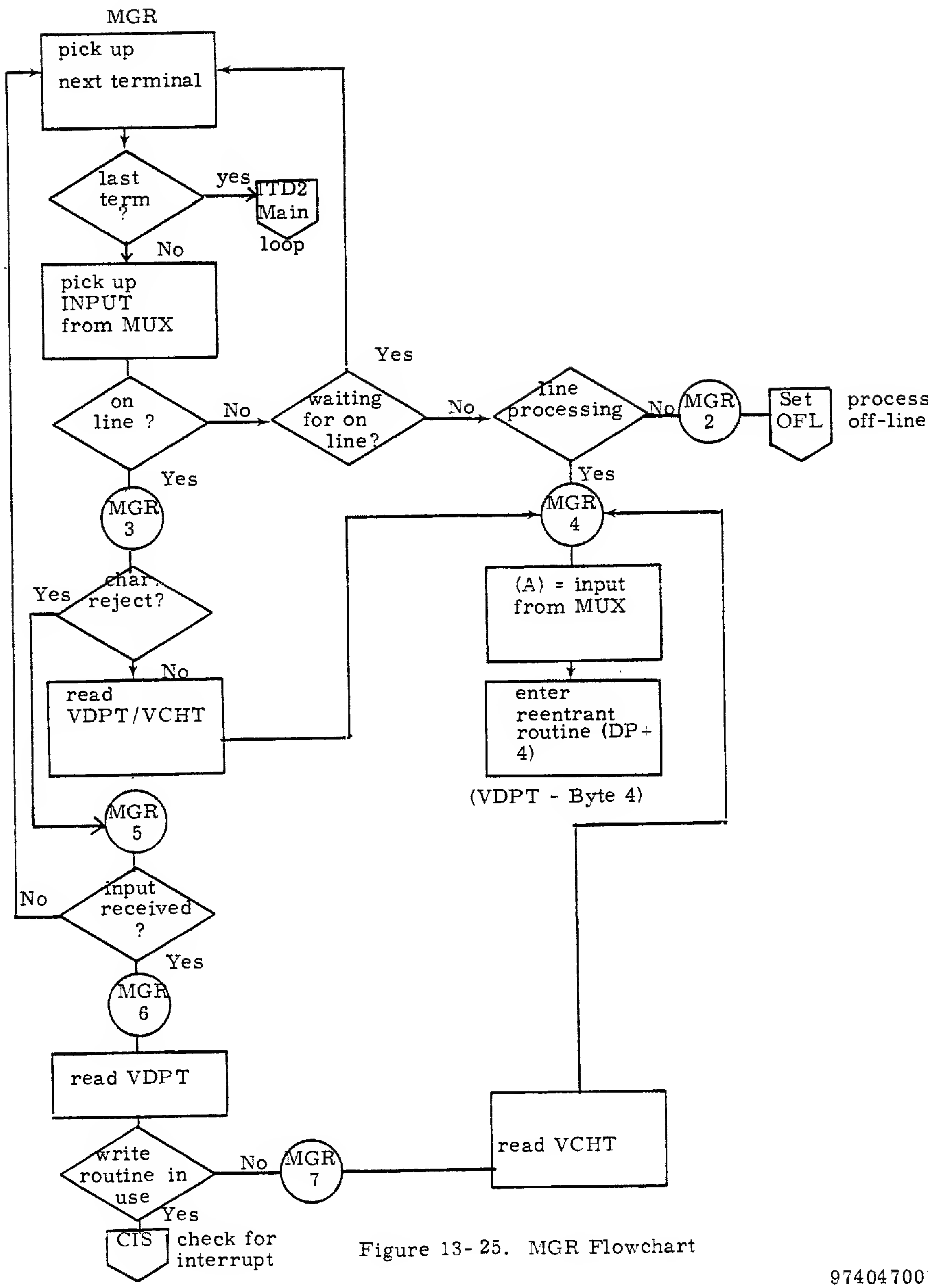


Figure 13-25. MGR Flowchart

Note: A line with two headed arrows
represents a subroutine
entered via RJM instruction

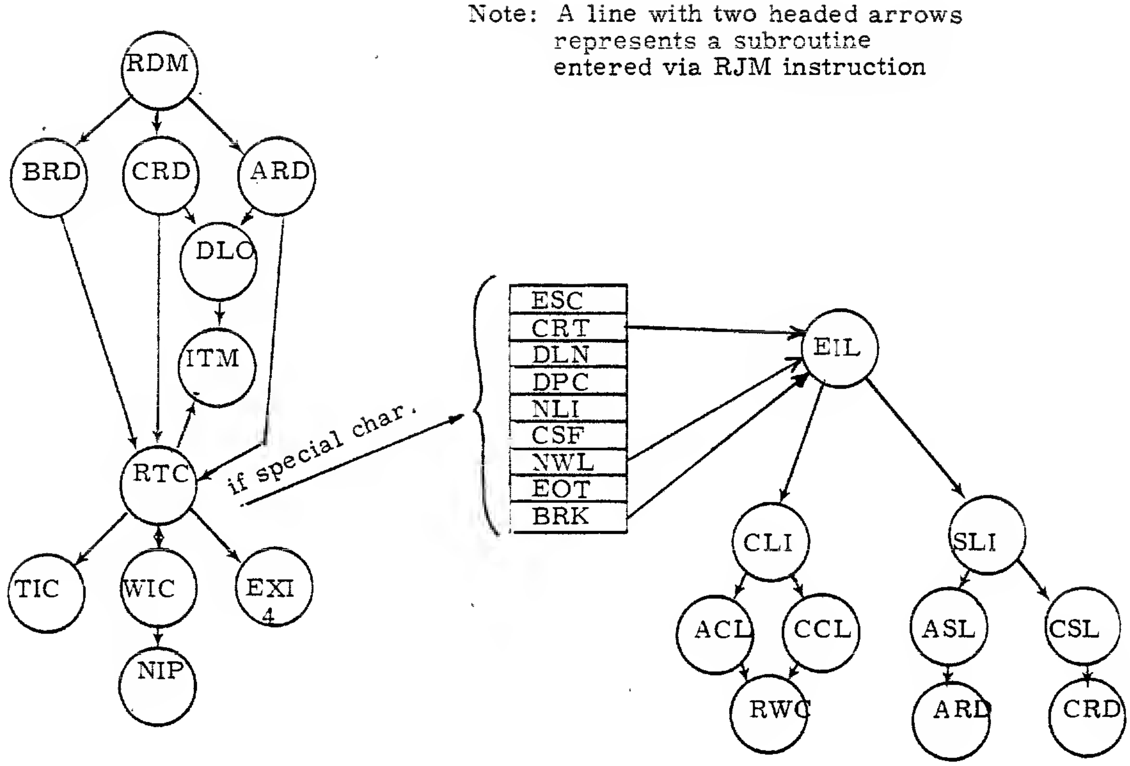
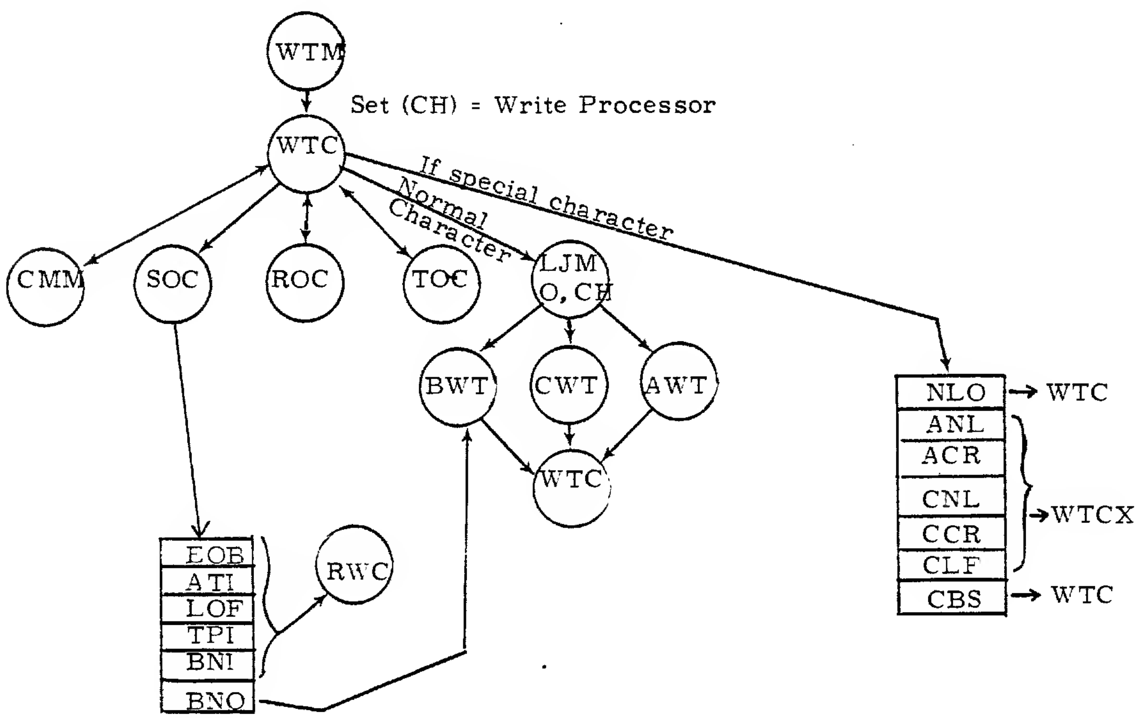


Figure 13-26. Read Mode Processing Subroutines.

TABLE 13-8. CONTROL SUBROUTINES

Control Byte	Subroutine Name	Function
0000	EOB	end of line
0001	EOB	end of block
0002	EOB	end of block
0003	ATI	AUTO input
0004	LOF	log off user
0005	TPI	set transparent input (allows all characters to be transmitted to the CPU program)
0006	BNI	set binary input
0007	BNO	begin binary output

The relationship between the write mode subroutine is shown in Figure 13-27.



Note: A line with two arrows indicate a return jump.

Figure 13-27. Write Mode Processing Subroutines.

The symbol qualifier PTP contains the routines used to process polled lines. These include:

SPL	-	sense polled lines
RPR	-	read poll response
PTR	-	process terminal response
SSC	-	set sequence count

Utility subroutines are under the symbol qualifier 1TD and are general subroutines used by the other routines described previously. The utility subroutines are as follows:

BUP	-	back up pointers
CUT	-	clean up terminal tables
ERQ	-	put entry in TELEX's request queue
RLT	-	read link table to get next pot in chain
RPC	-	read previous character in pot
SCA	-	set control address (for instance, RDM uses this to set read routine BRD, CRD, or ARD depending on translation table)
WTO	-	wait time out

Exit processing routines include:

MXE	-	process multiplexer error
DRP	-	process driver exit (call RESIDENT code set up by 1TD at initialization time)

13.6 1TA - TELEX AUXILIARY ROUTINE

1TA processes functions for TELEX which require PP action. The functions allowed are listed in Table 13-9.

TABLE 13-9. PROCESS FUNCTIONS

Overlay Name	Function Code	Routine Name	Description
	1		Unused
1TA	2	PFS	Purge file space
3TA	3	TFL	Adjust TELEX field length
3TB	4	RTJ	Return terminal job
3TC	5	CRF	Create rollout file - login
3TD	6	TLP	Terminal logout processor
3TF	7	FLS	Get file length in sectors Invoked via LENGTH command
3TF	10	SFD	Secondary file descriptions Invoked via STATUS, F
3TG	11	TIM	Time status command
	12		Unused
3TG	13	TIM	Increment time limit
3TL	14	IPF	Initiate primary file
3TH	15	RFP	Recovery file processor
3TI	16	SJS	Schedule SORT job
3TJ	17	GST	Gather terminal statistics
3TK	20	CUS	Clean up SALVARE file
3TM	21	CJS	Check job status

TELEX calls 1TA in one of two ways shown in Figure 13-28.

Group Request – A group of requests are stored in POTs. The input register format is:

59 42	41 36	35 30	29 12	11 0
1 T A	CP	0	Return Address	POT Pointer
18	6	6	18	12
IR IR+1	IR+2		IR+3	IR+4

where:
Return Address = Upper 24 bits of the word specified are set to zero upon completion of all requests.
CP = Control Point Number
POT Pointer = POT containing the list of requests
The requests are one word each with the following format:

59 36	35 24	23 12	11 0
Unused	FC	TN	ARG
12	12	12	12

where:
FC = function code
TN = terminal number
ARG = POT pointer or request type
The list of requests is terminated with a zero word.

Single Request - A single request is denoted by setting bit 2^{35} in the input register which is formatted:

1 T A	CP	4000B + FC	TN	ARG
IR IR1		IR2	IR3	IR4

where:
CP = Control point number
FC = Function code
TN = terminal number
ARG = pot pointer or parameter (depending on function)

Figure 13-28. TELEX Calls To 1TA.

1TA uses several bits in VROT of the terminal table. These bits are:

Bit	Description
0	Completion status bit
4	Set to indicate recall function by TELEX
10	Purge rollout FNT's
11	Error return

Figure 13-29 is the flowcharts of the initialization, execution, and termination of the control loop for 1TA.

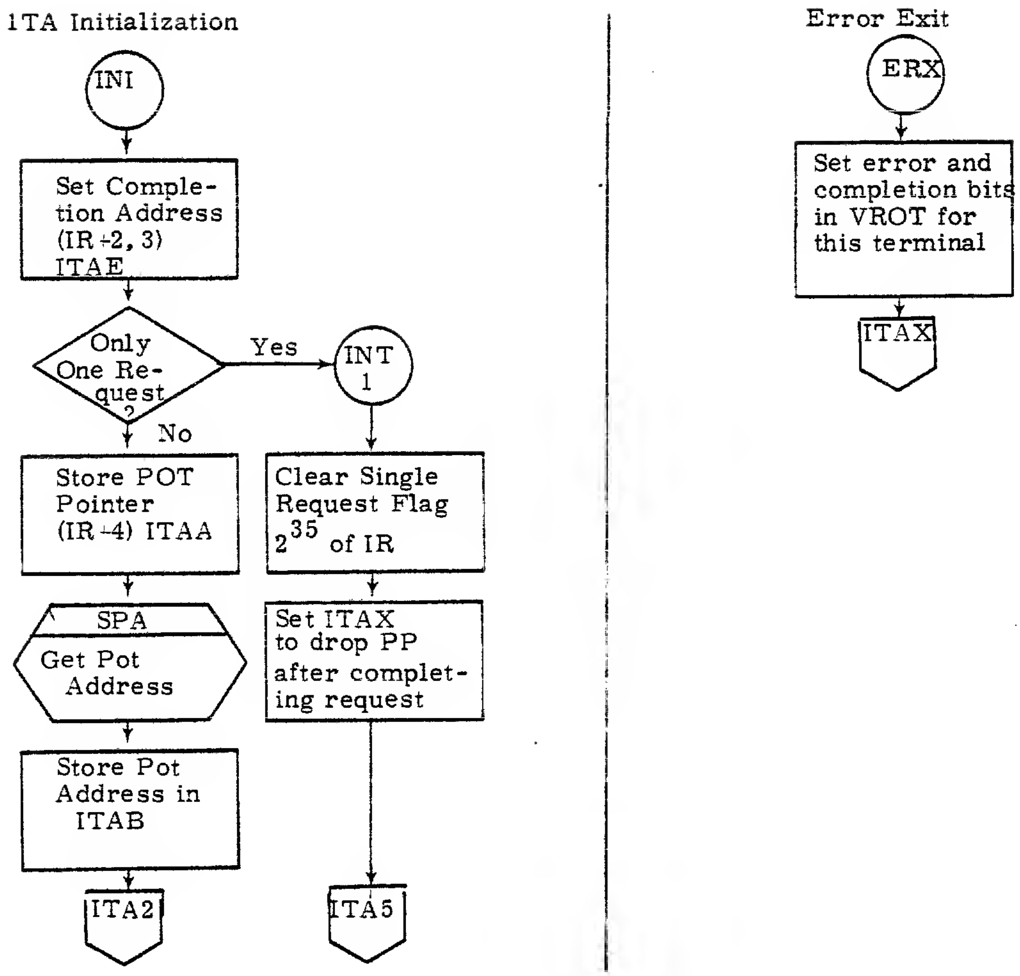


Figure 13-29. ITA Control Loop

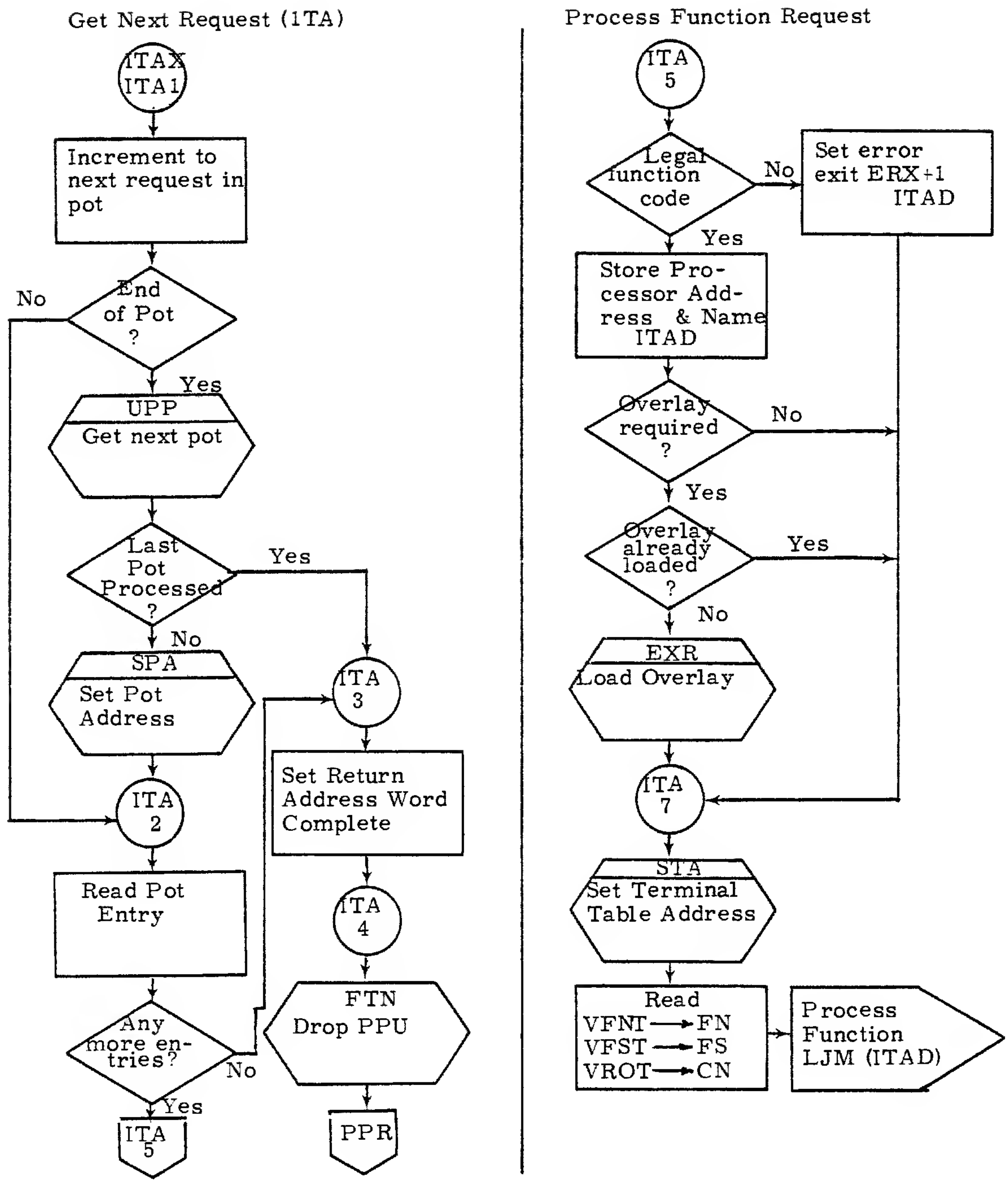


Figure 13-29. 1TA Control Loop (continued)

1TA Termination Routine - TER

Entry - (FS - FS+4) = Primary File FST
(CN - CN+4) = Rollout File FST
FP = First Pot of Message or data
LP = Last Pot of Message or data

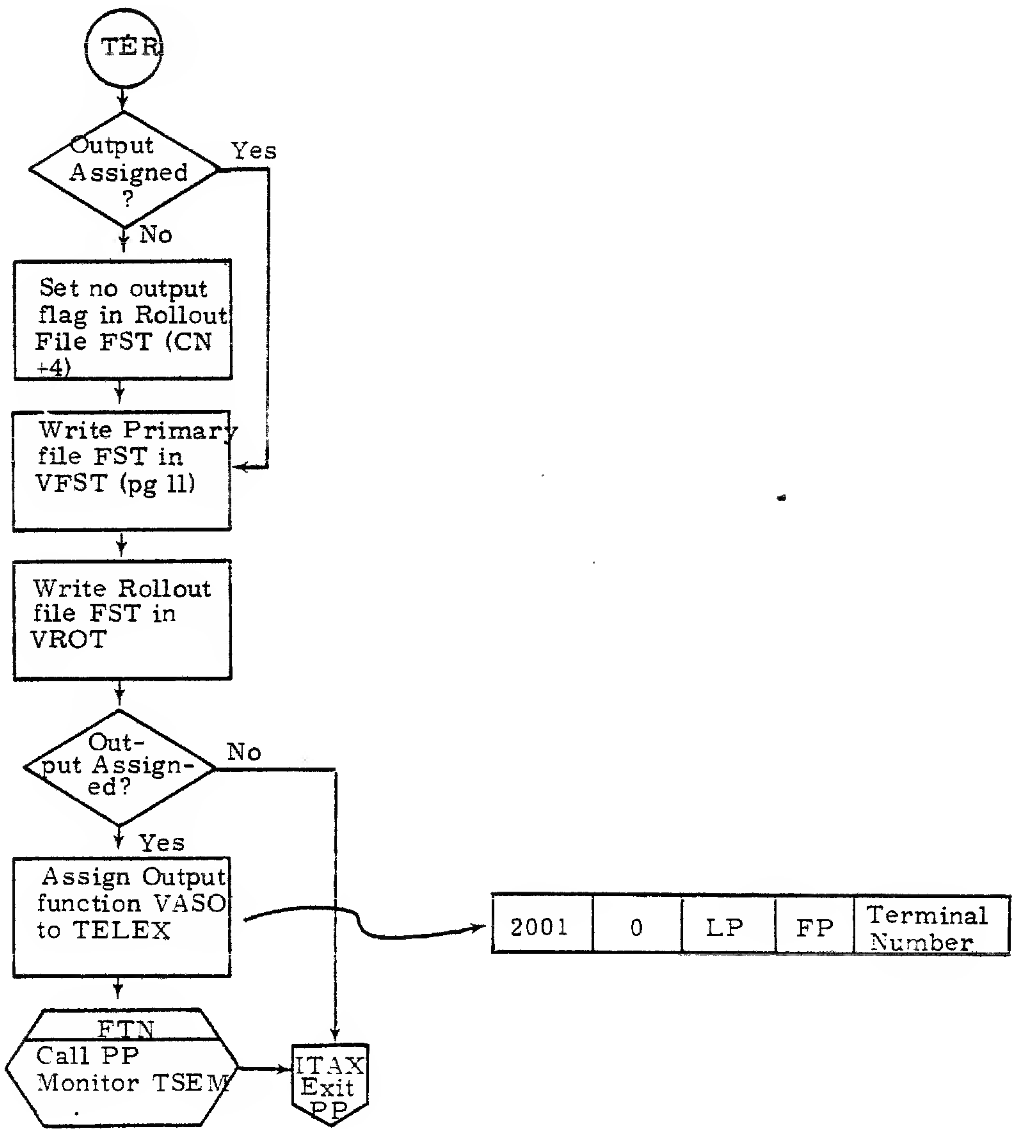


Figure 13-29. 1TA Control Loop (continued)

Function 5 is used to create a rollout file for a time-sharing job. The format of the rollout file is given in Figure 13-30.

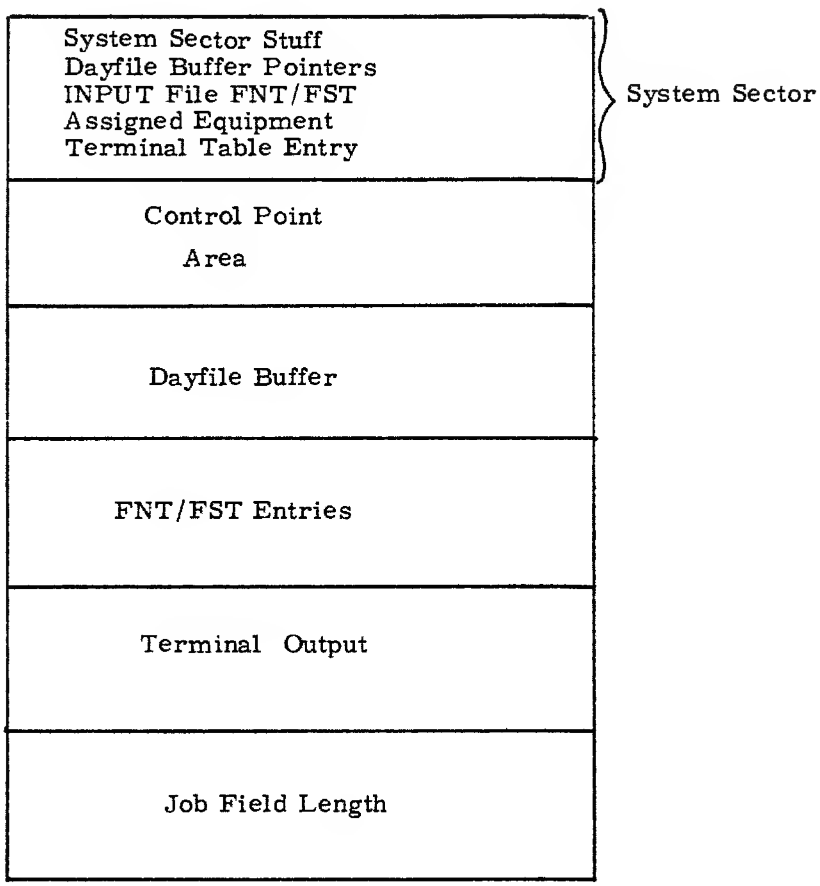


Figure 13-30. Time-sharing Job Rollout File

13.7 1TO - TTY INPUT/OUTPUT ROUTINE

1TO is called by TELEX to process a queue of requests for terminal input and output which require disk accesses. The queue resides in POTs within TELEX's field length The queue has been sorted by TELEX in order of equipment and disk addresses so as to minimize disk time. If there are requests for more than one mass storage device, the entries are processed for the first device available.

1TO is also called by 1RO to handle the first buffer of data on a rollout file. This data is passed to 1TO in a PP buffer. 1TO dumps the PP buffer into POTs and makes a VASO request to TELEX for that terminal.

The input register format when 1TO is called by 1RO as shown in Figure 13-31.

	IR	IR+1	IR+2	IR+3	IR+4
	59 42	41 36	35 24	23 12	11 0
(IR)=	1 T O	CP	0	BA	TN
	18	6	12	12	12

where:

CP = TELEX control point number
BA = Buffer address in PP of first sector of output data
TN = terminal number

Figure 13-31. Input Register -1RO

The input register when called by TELEX as shown in Figure 13-32.

IR	IR+1	IR+2	IR+3	IR+4
59 42	41 36	35 24	23 17	0
1 T O	CP	PP	0	Return Address
18	6	12	6	18

where:

CP = TELEX control point
PP = POT pointer to first POT of requests
Return Address = location of completion status word

Figure 13-32. Input Register - TELEX

The request in POTs are one word entries with the format shown in Figure 13-33.

59–54	53–48	47–36	35–33	32–27	26–24	23–12	11–0
RC	EQ	TRACK	W	0	X	FP	TN
6	6	12	3	6	3	12	12

where:

RC	=	Request code 0 = Correction dump 1 = Output data
EQ	=	Equipment number
TRACK	=	first track of file if RC = 0
	=	current track if RC = 1
W	=	number of words in last POT (0 means 10) W is meaninfgul when RC = 0.
X	=	number of POTs to dump. RC = 0
FP	=	first pot of source or output
TN	=	terminal number

Figure 13-33. POTs Entries

As a group of requests is completed, the above entries are updated by setting byte 2 to the last POT to be dropped or assigned These requests are then written back in the same pot from which they came.

The flowcharts of 1TO (Figure 13-34) show that it is broken down logically into 4 sections:

- Preset or initialization
- Main loop - get next request
- ICH subroutines = correction handler if RC = 0
- PRO subroutines to process output if RC = 1. That is data flow is:

  DISK→POTs → TTY

1TO Initialization - PRS

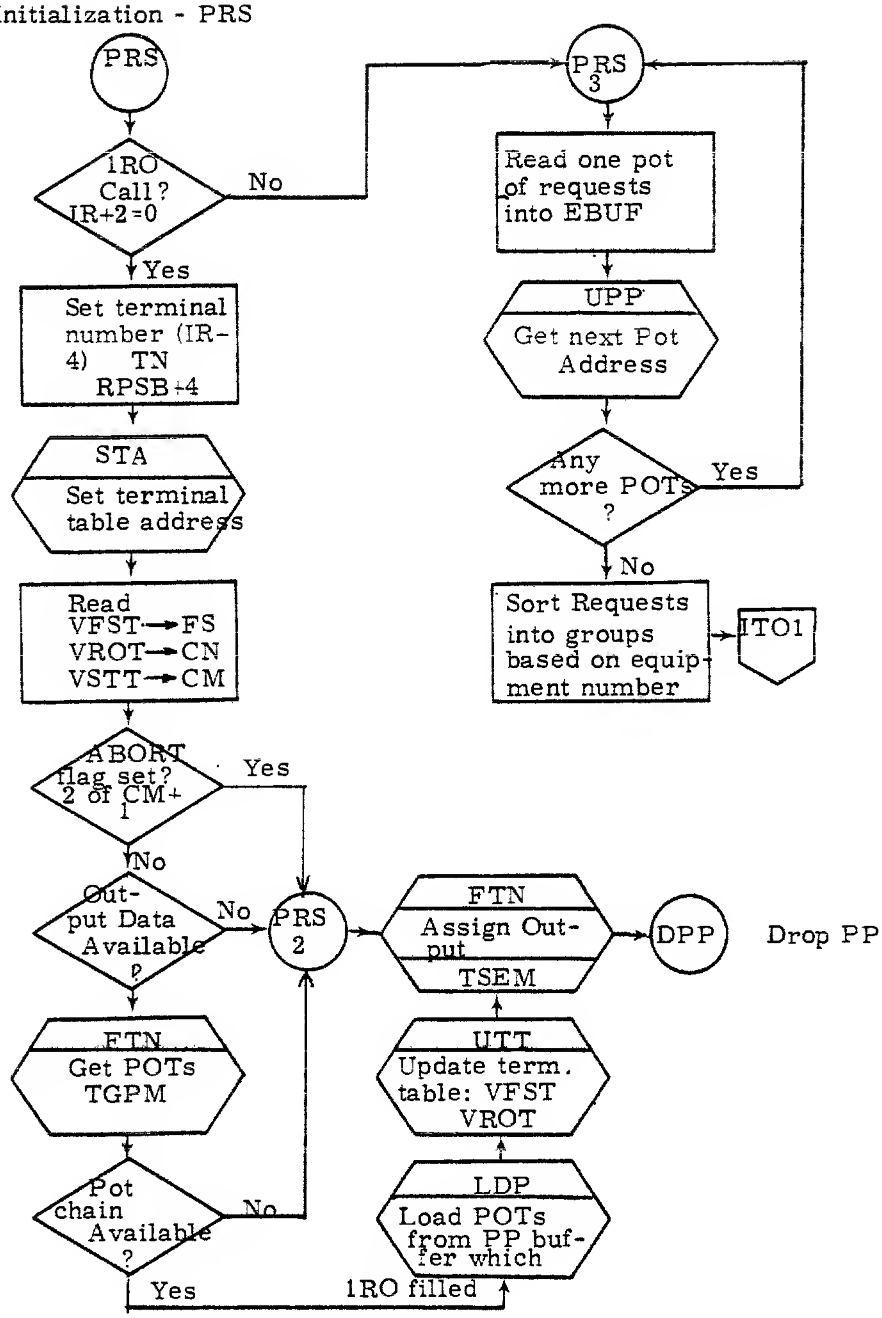


Figure 13-34. 1TO Flowchart

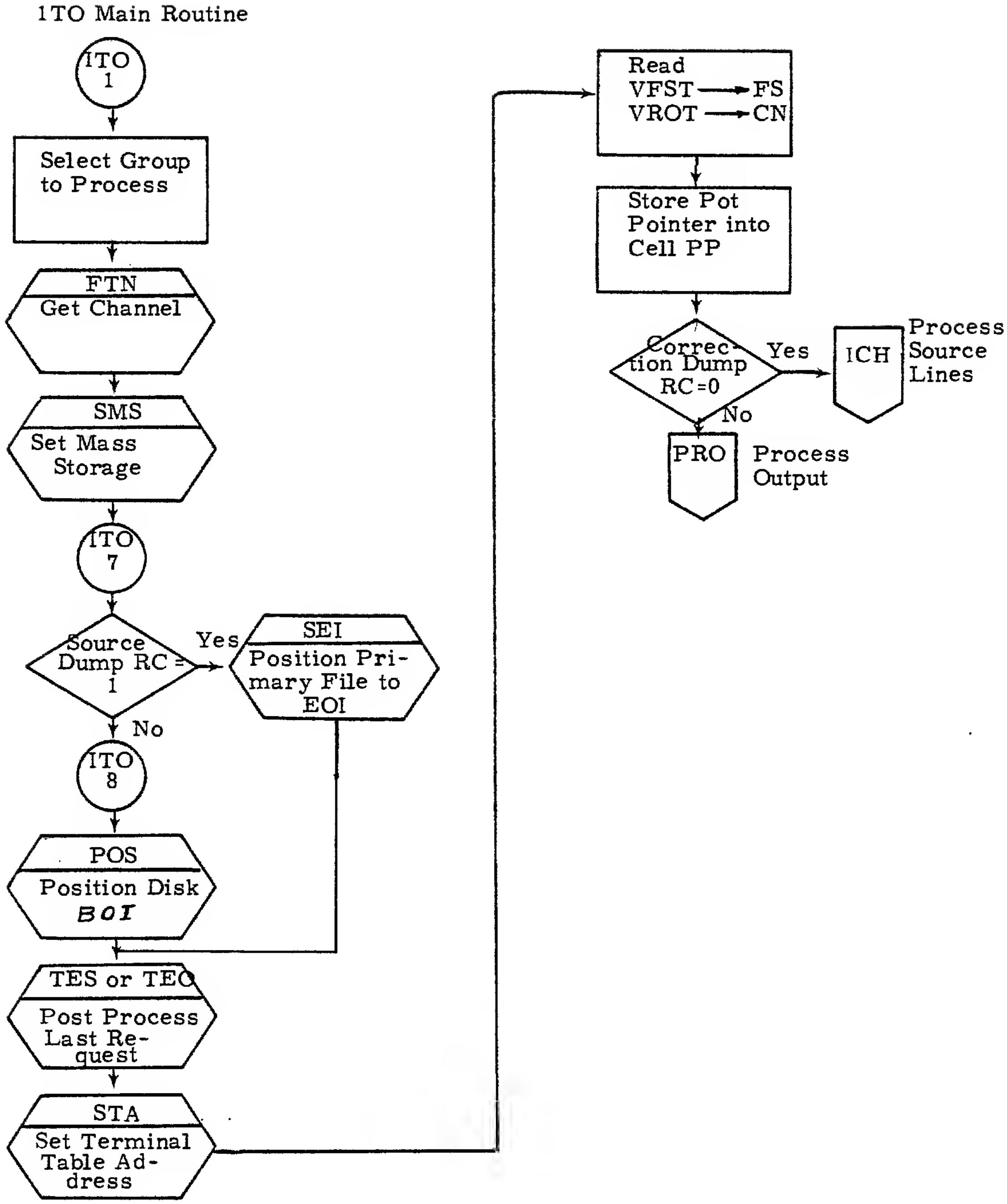


Figure 13-34. 1TO Flowchart (continued)

ITO10 - Common Return Point from ICH and PRO

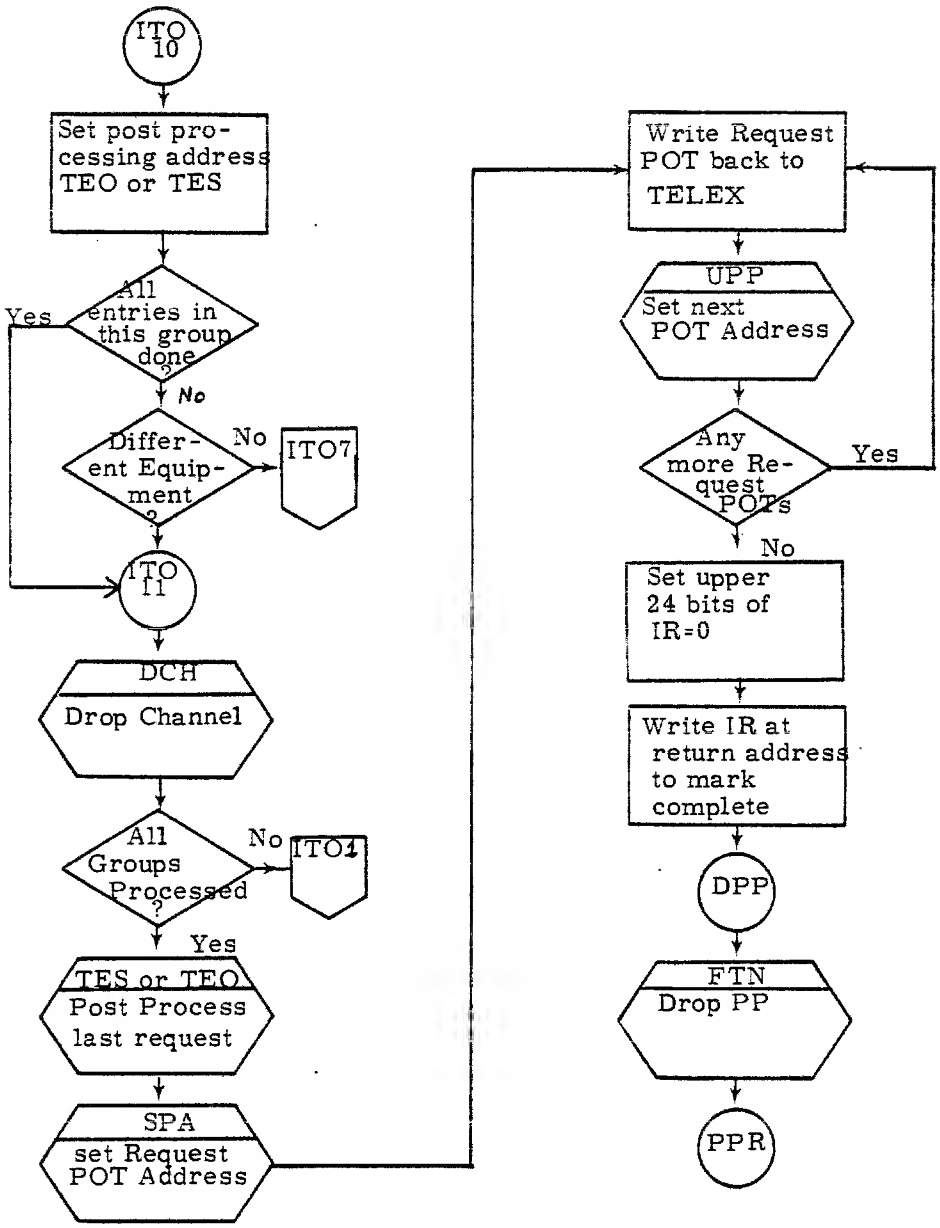
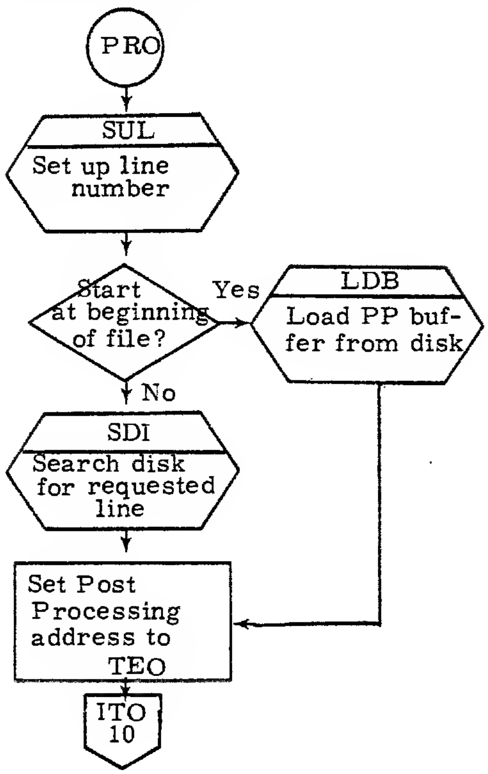


Figure 13-34. ITO Flowchart (continued)

PRO - Process Output
Disk→POTs→TTY



TEO - Terminate Output

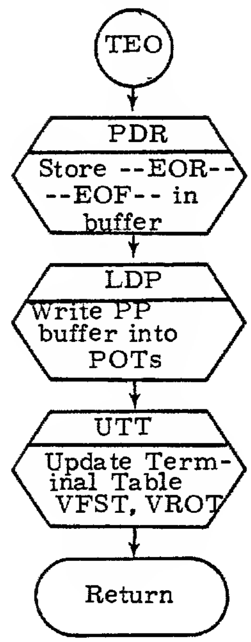


Figure 13-34. ITO Flowchart (continued)

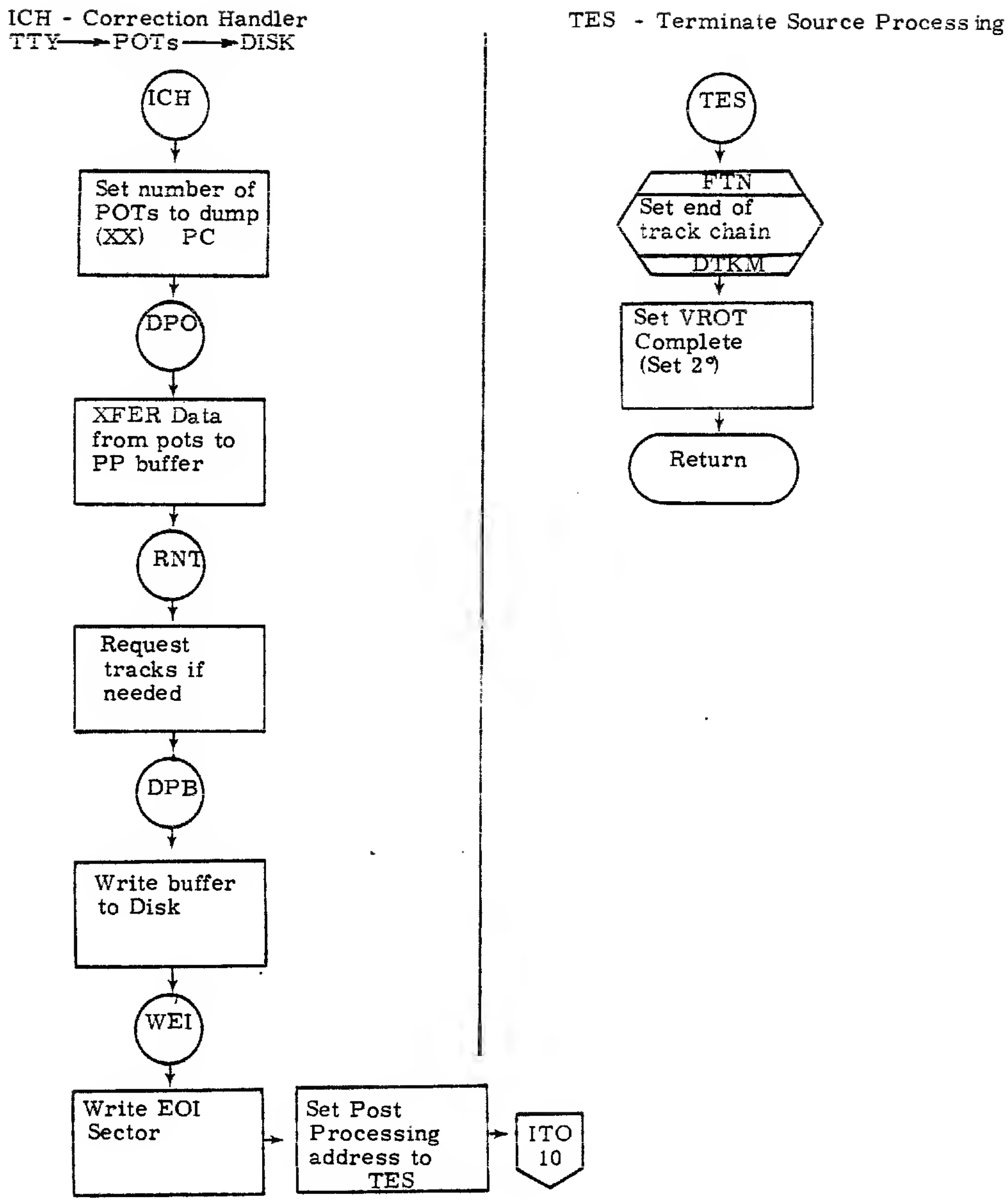


Figure 13-34. ITO Flowchart (continued)

13.8 SOME QUESTIONS AND ANSWERS ABOUT TELEX

Q. How does 1TD know what parity and transmission code a port is using before LOGIN has completed?

A. The NETWORK or SIMFILE file specified all the known terminals. In lieu of one of the above, 1TD assumes all terminals are 100 baud (10 cps) time sharing TTY type.

Q. What is $LDC issued by TELEX?

A. The "$" implies no local file load, this is a compiler call card issued by TELEX in response to a TTY user typing RUN or some similar call in the BASIC and other subsystems. There is no routine in the CLD or PLD named LDC. It is processed locally by 1AJ, who determines which compiler is requested and sets up the input file for it. The call is documented in the code and may be issued by any user. The subroutine CCL in TCS processes this card. The user may define a local file as LDC and load it if no "$" preceeds the call.

Q. What is the TT entry in VALIDUX used for and why?

A. The entry is used for validation. If the entry is set, then this user must be on that type of device to be validated. The TERMINAL table defines what type of terminal is calling.

13.9 SALVARE-TELEX RECOVERY FILE

The SALVARE file is built during TELEX initialization time. VPST is set = 4, the number of pseudo terminals and VMNL is set to the active number of users, determined from inspection of the MUX entries by 1TD.

During the INI21 code, the SALVARE file is built consisting of a two word entry per terminal. One word is needed per terminal so the file allows up to two users per terminal to be recovered. The situation can arise if the user at port x somehow is lost. Then a different user logs in at port x and is subsequently lost. The SALVARE file can be used to recover both of these users at a subsequent log in even at different ports.

If this is a recovery, TELEX will check the SALVARE file and insure that the number of ports has not changed since the original start up of TELEX. If there is a change, TELEX will abort. This code is at INI25.

During operation in TELEX1, the main loop (see Figure 13-6) will call CSF. CSF will issue a 1TA queue call to check the SALVARE file in 1TA routine CUS function 20. CUS will clear all entries in the SALVARE file over 10 minutes old. This call is made about every 3 minutes.

1TA is a combination of functions to perform for TELEX. The important functions associated with the SALVARE are:

1. CUS - clean up file
2. TLP - terminal log out processor
3. TRP - terminal recovery processor. This overlay contains the SALVARE format documentation.
4. RFP - recovery file processor

Since the SALVARE file is checked about every 3 minutes and entries more than 10 minutes old are eliminated, then:

1. A user that wishes to be recovered after losing contact must attempt recovery within 10 minutes.
2. In case of total system failure for over 10 minutes, any user wishing recovery after the system is revived must recover within 3 minutes. Otherwise, TELEX will clean up all entries, i.e., eliminate all entries within 3 minutes of TELEX recovery.

Recovery is accomplished in the routine RFP and a description of recovery processes follow:

OVERLAY (Recovery File Processor.)

RFP - Recovery File Processor call is:

	24	12	12	12	
IR=	1TA	15	TN	POT	at call time

Upon entry, IR+4 contains the parameter pot number. The pot contains the terminal table. IR+4 is set to the previous terminal number, which is recovered from parameter pot.

	24	12	12	12
IR=	1TA	15	TNN	TNB

TNN = Terminal number now.

TNB = Terminal number before.

A. To recover a user the entry on the SALVARE file is found and information returned to terminal table. The entry in the SALVARE file is cleared and the current rollout file is released. A dayfile message is issued saying user recovered.

B. A completion logout will be done for all entries that have been there longer than 10 minutes. At that time the files will be released and subsequent dayfile messages issued.

C. If necessary, the beginning and EOI sectors for each file will be validated to see if the users files are all there.

D. The status at the time the user was recovery processed is returned in VFST+4. (VROT+4) is returned as 0003.

The SALVARE file is always at FNT ordinal 1. It is initialized and set busy by 1TA. If 1TA finds the file active or mashed up (unrecognizable at recovery time) it will hang with MXFN monitor function. The format for the file is:

6	6	12	6	6	6	18
0	eq	FT	HRS	MIN	SEC	UI

eq = est ord of rollout file

FT = first track of rollout file

HRS.MIN.SEC = last entry time in compressed format

UI = User index

As an example, see the following dump of the SALVARE file. TELEX was active with 20 ports defined in the MUX entry. Only one user was active, and there were four pseudo terminals. The TTY responded TTY 4 to this user at log in, and the T display showed this user at terminal 4. The file consists of Sector 0, the system sector, section 1, the terminal recovery data, and sector 2, the EOI.

Sector 1 data is 50 words long, with two entries for each terminal which are pseudo terminals 0 thru 3 and actual terminals 4 thru 23.

The recovery information for the user at Terminal 4 is:

eq = 0

FT = 557

time = 24.45.43

UI = 1.

DUMPTK(TK=302)   DUMPTK - VER. 1   74/07/21. 20.41.33.   PAGE 1

WORD	TK=302 SE=0 B1=3777 B2=77		TK=302 SE=1 B1=2 B2=50		TK=302 SE=2 B1=0 B2=0 EDI	
0	23011426012205000701	SALVARE GA	00000000000000000000	0 Pseudo Ports	00000004053500503242	OE2 /Z7
1	00004302007601003305	B0 ¬A OE	00000000000000000000		01532301142601220500	A$SALVARE
2	00000000040745242023	D0UIPS	00000000000000000000	1	00240002000300000311	T B C CI
3	30776370114417016040	X‡‡↑1VOA≡5	00000000000000000000		17010002000404560307	OA B DD:CG
4	30261207230010775400	XVJ≤5 H‡=	00000000000000000000	2	05003305000404650004	E OE DO* D
5	71261405600150034201	↓VLE≡AACZA	00000000000000000000		05350004046500041466	E2 DO* OLV
6	34023040051554007076	1GX5EM= ↑¬	00000000000000000000	3	00010100100000030200	AA H CB
7	30431277310160033405	X6J‡YA≡C1E	00000000000000000000		30703071307214771701	X↑X↓X<L‡OA
10	30071003160690460313	XGH≤NF≡5CK	00004557244543000001	4 ↑.T↑B A	05763075605030511237	E¬X2≡/X(JA
11	14045400707637020603	LU= ↑¬4BFC	00000000000000000000		10073474305113771006	HG1$X(K‡HF
12	01007003310160033405	A ↑≤YA≡C1E	00000000000000000000	5	33500462100602000135	O/D1HFB A2
13	30031071125111500564	XCH↓J(1/E≠	00000000000000000000		02000424011500055400	B OTAM E=
14	30061000761400071003	XFHE6(XGHC	00000000000000000000	6	01333001101431026114	AOXAHLVB(L
15	16056010301012342300	NE≡¬XHJ1S	00000000000000000000		15420100053610145400	H7A E3HL=
16	00000550300710031606	E/XGHCNF	00000000000000000000	7	03261063230023005400	CVH1S S =
17	60103040331005413041	≡HX>OHE6X6	00000000000000000000		03252001301402000335	CUPAXLB C2
20	33110520304253120515	U1E¬X/OJEM	00000000000000000000	10	05522002140102000335	E1PBLAB C2
21	30433313137705115000	X0OAK‡E1/	00000000000000000000		05041464617077745400	EDL≠(↑‡S=
22	70760510301404041070	↑¬E¬XLODH↑	00000000000000000000	11	02721702341750000533	B<DB1O/ ED
23	12010503010070503014	JAE≤A ↑/XL	00000000000000000000		54000275500001355400	= B2/ A2=
24	34033007100316046010	1CXVHCNDEH	00000000000000000000	12	03015000032554000305	CA/ CU= CE
25	37135400714730261074	4K= ↓↑XVHS	00000000000000000000		50000326540003061422	/ CV= CFLR
26	22000017160134043111	R UNA1UY1	00000000000000000000	13	02000364307660030200	B C≠X¬≡CB
27	34063003100607333011	1FX≤HFG0X1	00000000000000000000		05475600057603132000	E*, E¬CKP
30	34063007100316203272	1FX≤HCNPZ<	00000000000000000000	14	04773517300710060607	D‡2DXGHFFG
31	54007177100343002100	= ↓‡H1S Q	00000000000000000000		10713406140034070200	H↓1FL 1GB
32	54007176300610752100	= ↓¬XFH2G	00000000000000000000	15	06064017341650170001	FF5D1N/O A
33	33206010300612033414	UP≡¬XFJC1L	00000000000000000000		34033017020006161057	‡CXDB FNH.
34	50140010340637040563	/L ¬1F4UE‡	00000000000000000000	16	34014017340730164417	‡A5D1GXN90
35	30053201340540070100	AEZ¬1EXGA	00000000000000000000		36174017107110213303	3D5DH↓H0OC
36	70030100721754007231	↑CA <U= <Y	00000000000000000000	17	44170605300105030100	9OFEXAECA
37	14056001300322013402	LE≡¬XCZA1B	00000000000000000000		02213004020004462000	BQXDB O-P
40	20007230310160033405	P <AYA≡C1E	00000000000000000000	20	15273415200011115400	HW1HP I1=
41	30071003160601707321	XGH≤NF[↑>U	00000000000000000000		05332000053654000135	EOP E3= A2
42	50007324137733105400	/ >‡K10H=	00000000000000000000	21	14002302361401010134	L SB3LAAA1
43	73246306370207443101	>TC≤4BG9YA	00000000000000000000		30143502106335011014	XL2BH12AHL
44	60033405300310711251	=C1≡XCH↓J(	00000000000000000000	22	31026010301113771006	VB≡HXIK‡HF
45	11500506300610060763	1/EVXFHFG‡	00000000000000000000		33100410230435030561	0HOH5D2CE(
46	30071003160560103010	XGH≤NE≡HXH	00000000000000000000	23	30120502301534150100	XJEBXM1MA
47	12340554300710031606	J1E≠XGHCNF	00000000000000000000		01525400031110636010	A1= C1H‡≡H
50	60101400340450047321	=HL 1D/D>U	60101400340450047321	≡HL 1D/D>U	30113402301034010346	X11BXH1AC-
51	53040010054136041104	$D HE63D1D	53040010054136041104	$D HE63D1D	00000001163500000000	AN2
52	05703005320134053007	E↑XEZA1EXG	05703005320134053007	E↑XEZA1EXG	14771701057630766010	(‡OAE¬X¬≡H
53	01007217000000000000	A <D	01007217000000000000	A <D	30100571010004443410	XHE↓A D91H
54	00000000001007326200	A >VP	00000000001007326200	A >VP	30766210301017360761	X¬1HXH03G(
55	11130200131450001100	1KB KL/ 1	11130200131450001100	1KB KL/ 1	20013113601030103111	PAYK≡HXHY1
56	05100200120316020320	EHB JCNBCP	05100200120316020320	EHB JCNBCP	05722001310563700350	E<PAYE‡↑C/
57	14020100131002006305	LBA KHB ‡E	14020100131002006305	LBA KHB ‡E	20013105261060103014	PAYEVH≡HXL
60	04720200635004061400	U<B ‡/DFL	04720200635004061400	U<B ‡/DFL	04410371307416206010	D6C↓X$NP≡H
61	34571444010013103057	1.LYA KHX.	34571444010013103057	1.LVA KHX.	30143456301334550100	XL1.XK1 A
62	60201701601056001312	=POA≡H, KJ	60201701601056001312	=POA≡H, KJ	01231457601030743210	ASL.≡HX$2H
63	30131204055130201277	AKJDE(XPJ‡	30131204055130201277	AKJDE(XPJ‡	05621411020003640356	E1L1B C≠C.
64	34055100055160103010	1E(E(≡HXH	34055100055160103010	1E(E(≡HXH	01001752341114120200	A O111LJB
65	10060704140601001310	HFGDLFA KH	10060704140601001310	HFGDLFA KH	03640371010002713411	C≠C↓A B↓1I
66	02000547301410031620	B E*XLHCNP	02000547301410031620	B E*XLHCNP	14040200036403712001	LDB C≠C↓PA
67	32725400460610632300	Z<= -FH1S	32725400460610632300	Z<= -FH1S	03706373000163120573	C↑1> A‡JE>

KRONOS TRANSACTION SUBSYSTEM 14

14.0 INTRODUCTION

Transaction processing is handled as a distinct subsystem within the KRONOS operating system; therefore, all of the features available under KRONOS are retained when KRONOS is operating as a transaction processing system. These features include:

- Local batch processing
- Remote batch processing
- Deferred batch processing
- Interactive terminal processing (also called application program)

Under the transaction subsystem, a user program is referred to as a task. A task is the absolute binary code generated from the assembly or compilation of the user program.

These tasks read and update information on the subscriber's data base and generate output to the transaction terminals. User programs reside at subcontrol points within the field length of the transaction executive (TRANEX), which resides at control point 2.

The subcontrol point feature allows the transaction executive to maintain complete control over each task. Some of the advantages associated with subcontrol points are:

1. Isolation of one subcontrol point from other subcontrol points and the transaction executive, guaranteeing system security.
2. Blocking of RA+1 requests from a subcontrol point. No PP requests or I/O actions are allowed directly from a subcontrol point. Any such requests are intercepted by the system monitor which returns control to the control point executive program.
3. Freedom to move, load, and overlay areas within the subsystem field length. Since each subcontrol point has a relative origin of zero, absolute overlays all originating at a given address (for example, 100B) can be loaded in any order and at any place within the subsystem field length.

The transaction executive allows a maximum of 31_{10} subcontrol points. An installation parameter sets the number of subcontrol points that the transaction executive initializes. See Section 3 of the KTS Reference Manual for a complete description of the transaction executive. When the transaction executive is loaded, the operator may select a number of subcontrol points other than this default value. The number of subcontrol points must not be less than two or greater than 32. Once the transaction executive is initialized, no change in the number of subcontrol points is allowed.

Each subcontrol point requires eight words of table space within the transaction executive. No space, other than a table entry, is allocated for a subcontrol point unless it is active. The optimum number of subcontrol points is selected by the site. It is suggested that 6 to 12 subcontrol points be used initially.

The KRONOS data manager controls the structure of user data, thereby relieving the user of this responsibility. In order to control this data, the data manager must be supplied information about a user, his application area, and installation. This information is provided by the user at data base definition time. A data base can be defined as this control information together with transaction data supplied by the user.

The transaction data consists of logically related data files. Data files have specific names that provide a common point of reference between user programs and the data manager. Data files are structured into logical groups of information called records. Records may be subdivided into elements. One or more elements may serve as a key or identifier for a record.

At data base definition time, the user supplies a description of all data elements and data files to be contained in the data base. Changes to the data base definition can be made by using the data base utility DBFORM.

The information provided to the data manager consists of parameters that describe the physical allocation of the data, parameters that describe the element characteristics and security, and parameters that describe the file organization.

When accessing data through the data manager, user programs require only a minimum amount of information concerning the data. The data manager structures the data for rapid, efficient retrieval. The user program need not be aware of the structure of the data it accesses.

At data base definition time, the user supplies a description of all data elements and data files to reside in the data base. DATADEF converts this description to a file known as the Element Descriptor Table (EDT) for the data base. DBFORM is a utility which actually creates the data base according to the EDT. DBFORM can also be used later to modify the initial data base. When modifications are to be made, a new EDT is usually required.

Figure 14-1 shows the relationship between the three utilities DATADEF, DBFORM, and DATAMAP and batch job data manager DBMI. First, the User defines the data base structure. DATADEF (1) creates the Element Descriptor Table (EDT). DATAMAP (2) reads the EDT and produces a listing. Second, the User specifies filenames. DBFORM (3) appends those names to the appropriate EDTs. DBFORM preallocates files on disk, thus creating empty permanent files. TAPE is used when reformating existing files. The

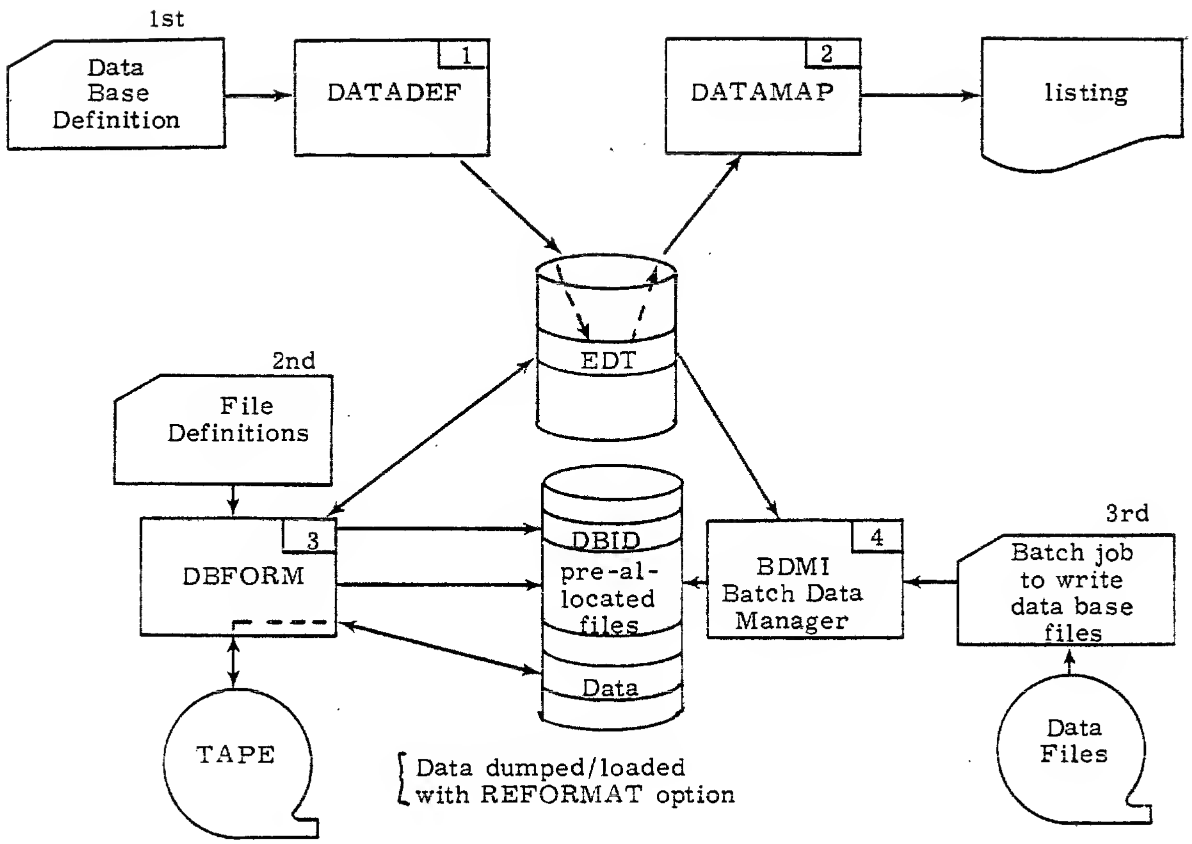


Figure 14-1. Data Base Creation

Data Base Identification file (DBID) is only used by TRANEX. Third, batch job writes data to files using data manager commands. BDMI (4) stores data in pre-defined files based on the EDT specifications.

For a more detailed description of DATADEF and DBFORM, consult the Transaction Subsystem Reference Manual.

14.1 KTSPL STRUCTURE

The KRONOS Transaction Subsystem (KTS) is a set of routines that provide transaction processing capabilities. All KTS routines are available on the KTSPL. The structure of the KTSPL is outlined as follows:

- Common decks
- PP routines
- CP routines
- User library routines
- Tasks

14.1.1 Common decks include the following:

COMBACM	-	Access methods
COMBACT	-	Add entry "CRAT" (Copy Reference Address Table) table
COMBBCT	-	Build a copied record address table
COMBDBM	-	KTS Data Manager
COMBELP	-	CYBERLOAN element processor
COMBINT	-	CYBERLOAN initialize data manager
COMBSCT	-	Search a copied record address table
COMKMAC	-	Data Manager Macro definitions
CALLKTS	-	A routine similar to CALLCPU that calls COMKMAC to obtain an individual listing of KTS interface macros.

14.1.2 There are only two PP routines associated with KTS. They are:

CS1	-	PP portion of KTS stimulator
1TP	-	An auxiliary PP routine call by TRANEX (via TLX) to process various functions

14.1.3 The CP routines include the following:

DBFORM	-	Utility to create/update DBID file
DATADEF	-	The 2-pass Data Definition Language (DDL) (COMPASS) compiler.
DATAMAP	-	A routine that provides a concise map of a data base description as established by DATADEF.
LIBTASK	-	The KTS utility used to build and edit a task library.
PRESIM	-	Converts data to TRANSIM's input format.
TRANSIM	-	Interface between user program and CS1.
TRANEX	-	The KRONOS Transaction Executive.

14.1.4 The user library routines include FORTRAN Extended and COBOL interface routines and the batch user's data manager interface. These routines written in CP COMPASS are:

BDMI	-	The interface necessary to execute the KTS Data Manager from a batch job.
CALLTSK	-	FORTRAN Extended subroutine to request scheduling of tasks.
CEASE	-	Terminate task execution.
CMDUMP	-	Dump a task's central memory, exchange package, and/or data manager buffers.
DMGR	-	KTS data manager interface routines (GETN, PUT, REPOS, etc.)
DSDUMP	-	Allows a task to change any of the default CM dump options.
JOURNL	-	Allows a task to write entries on the JOURNAL file.
SEND	-	Enables a task to send a message to a terminal.
TARO	-	Enables a task to alter the "user argument" area within a terminal table entry.
TSIM	-	Allows a task to get terminal status and information.

14.1.5 There are four tasks provided on the KTSPL. They can be modified and then stored on a task library permanent file under the user number and password assigned to the Transaction Subsystem. The four tasks are:

ITASK	-	An initial task used to interface between TRANEX and other user application tasks. It processes all transaction input to determine which user task to call. ITASK is provided to serve as an example and will vary for each installation.
KDIS	-	The TRANEX K-display command directory. KDIS is a task initiated by the operator by typing K.SWITCH. The purpose of the task is to save core in TRANEX.
MSABT	-	A system task which sends error messages to the originating terminal when a transaction ends abnormally.
OFFTASK	-	A task scheduled by TRANEX when a request is made for an inactive task (i.e., a task in the task library but has been turned off either by LIBTASK directive or operator command). OFFTASK simply sends a message to the originating terminal informing it that an inactive task was requested.

14.1.6 Figure 14-2 is two pages from a KRONREF run using the KTSPL. Only the cross reference of the common decks called is shown. The cross reference of symbols used is not shown since there are very few references with only two PP routines.

14.2 TRANEX

The relationship between TRANEX and TELEX is shown in Figure 14-3. Notice that the Time-Sharing Executive (TELEX) runs at control point 1, while the Transaction Executive (TRANEX) runs at control point 2. This is done to avoid a storage move of the two executives which would be necessary if they resided at other control points. Transactions

CROSS REFERENCE OF OPL. OPL FILE=KTSPL SYS. TEXT=SYSTEXT (KRONOS 2.1-01/AB) 73/09/25. 09.52.24. PAGE 15

COMMON DECK CALLS.

DECK	DECK REFERENCES.								
COMBACM	DBFORM	TRANEX	BDMI						
COMBACT	DBFORM	TRANEX	BDMI						
COMBBCT	DBFORM	TRANEX	BDMI						
COMBDBM	TRANEX	BDMI							
COMBELP	DBFORM	TRANEX	BDMI						
COMBINT	TRANEX	BDMI							
COMBSCT	DBFORM	TRANEX	BDMI						
COMCARG	DATADEF	DATAMAP	DBFORM	LIBTASK	PRESIM	TRANEX	TRANSIM		
COMCCDD	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI
COMCCFD	DATADEF	TRANEX							
COMCCIO	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI
COMCCOD	DATADEF	KTSDMP	PRESIM	TRANEX	BDMI				
COMCCPM	DBFORM	KTSDMP	LIBTASK	TRANEX	TRANSIM	BDMI			
COMCDXB	DATADEF	DBFORM	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI		
COMCEDT	TRANSIM								
COMCLFM	DBFORM	PRESIM	TRANEX	TRANSIM					
COMCMAC	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI
COMCMTM	DATADEF	DATAMAP	DBFORM	LIBTASK					
COMCMTP	DATADEF	DATAMAP	DBFORM	LIBTASK					
COMCMVE	DATADEF	DATAMAP	DBFORM	LIBTASK	TRANEX	BDMI			
COMCOVL	TRANEX								
COMCPFM	DBFORM	KTSDMP	LIBTASK	TRANEX	BDMI				
COMCRDC	DBFORM	PRESIM	TRANEX	TRANSIM					
COMCRDO	DATADEF	DATAMAP	DBFORM						
COMCRDS	DATADEF	DBFORM	LIBTASK	PRESIM	TRANEX				
COMCROW	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI
COMCRTN	PRESIM	TRANEX							
COMCSFM	TRANEX								

Figure 14-2. KRONREF Run

CROSS REFERENCE OF OPL. OPL FILE=KTSPL SYS. TEXT=SYSTEXT (KRONOS 2.1-01/AB) 73/09/25. 09.52.24. PAGE 16

COMMON DECK CALLS.

DECK	DECK REFERENCES.										
COMCSFN	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	TRANEX	BDMI				
COMCSRT	LIBTASK										
COMCSST	LIBTASK										
COMCSYS	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI		
COMCUPC	LIBTASK	TRANEX	BDMI								
COMCWOD	KTSDMP	BDMI									
COMCWTC	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANSIM	BDMI			
COMCWTO	DBFORM	BDMI									
COMCWTS	DATADEF	DBFORM	LIBTASK	PRESIM							
COMCWTW	DATADEF	DATAMAP	DBFORM	KTSDMP	LIBTASK	PRESIM	TRANEX	TRANSIM	BDMI		
COMKMAC	CMDUMP	DMGR	DSDUMP	JOURNL	SEND	TARO	TSIM	ITASK	KDIS	MSABT	OFFTASK
COMPC2D	CSI										
COMPMAC	CSI	ITP									
COMPRSI	ITP										
COMSLDR	BDMI										
COMSNET	PRESIM	TRANEX									
COMSPFM	DATADEF	DBFORM	KTSDMP	TRANEX	BDMI						
COMSPFU	DBFORM										
COMSREM	CSI										
COMSSSJ	TRANSIM										

Figure 14-2. KRONREF Run (Continued)

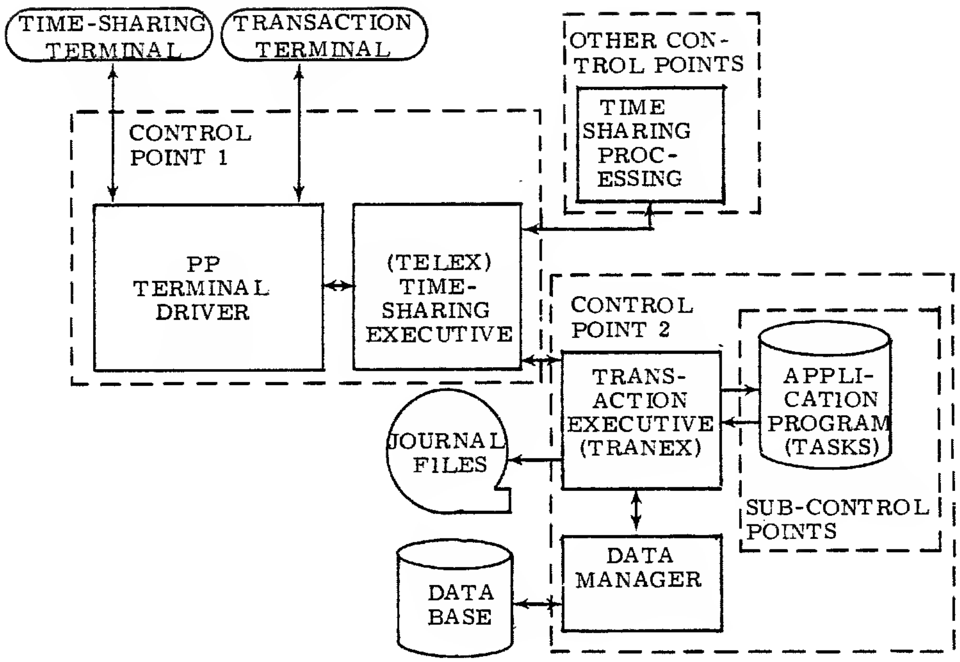


Figure 14-3. TRANEX-TELEX Relationship

are passed between TRANEX and TELEX via inter-control point communication. That is, the CPUMTR function, SIC, is used to transfer data between control points.

Figure 14-4 shows the breakdown of the TRANEX control point. The data manager code is contained in common deck COMBDBM and is called by TRANEX and by BDM1 for batch processing. TRANEX will support up to 31 subcontrol points.

Since time does not allow for a detailed level of documentation for the entire subsystem and associated utilities, only the following routines will be discussed.

- TRANEX – Control Point Initialization and Termination
- TRANEX1 – Executive Initialization
- TRANEX – Run Time Executive
- TRANEX2 – Recovery/End Processor

TRANSACTION CONTROL POINT 2

Area	Address
TRANEX	RA_2
DATA MANAGER	RA_2+5000_8
EXECUTIVE TABLES AND BUFFERS (TST, TLD, EDT, DM BUFFERS)	$RA_2+15400_8$
Sub-Control Point Area	$RA_{S1} - 100_8$
Initial Task	RA_{S1}
Free Core	
Sub-Control Point Area	$RA_{S2} - 100_8$
TASK	RA_{S2}
Free Core	
Sub-Control Point Area	$RA_{S3} - 100_8$
TASK	RA_{S3}
FREE CORE	
	$RA_2 + FL_2$

Figure 14-4. TRANEX Control Point

14.3 TRANEX – CONTROL POINT INITIALIZATION AND TERMINATION

TRANEX is initialized similar to TELEX. That is, 1TP is called by 1DS when the operator types: TRANEX. 1TP initializes the TRANEX control point by performing the following functions:

- Checks error flag
- Sets jobname "TRANEX" in control point area
- Sets CPU priority to 76
- Requests 50,000B words of CM
- Writes the control card buffer

The control card buffer contains the following:

```
TRANEX1.
TRANEX2.
EXIT.
TRANEX2.
```

When 1TP is dropped, 1AJ processes the next control card, TRANEX1, thus loading the CP initialization code for the TRANEX subsystem. TRANEX1 is absolute and loads at RA+101B. TRANEX1, after performing initialization, loads the run time program, TRANEX. The control card TRANEX2 will be processed whenever TRANEX is stopped by the operator or when an abnormal condition is detected by TRANEX. If TRANEX2 finds no errors and RECOVERY was selected (operator selected sense switch 4, ONSW4, at TRANEX control point), TRANEX2 restarts the subsystem by calling 1TP to perform initialization. (1TP is called via the TLX monitor function.) If the operator selected sense switch 5, ONSW5, TRANEX2 calls DMP to dump the TRANEX field length and then calls OUT (uses RELEASE macro, which calls LFM) to print the dump. In this case, if sense switch 4 (ONSW4) is also selected, restart is initiated as above. When looking at the dump, remember that TRANEX2 code was originated (ORG) at the K-display processing code (KDIS), and this portion of TRANEX is wiped out by TRANEX2; since both use the same memory area, similar to TELEX and TELEX1.

14.4 TRANEX1 – TRANSACTION EXECUTIVE INITIALIZATION

TRANEX1 is loaded by 1AJ in response to the control card call set up by 1TP. As an initialization routine, it builds tables, allocates buffers for TRANEX and the data manager, and attaches files for TRANEX and the data manager. TRANEX1 drives the K-Display to allow the operator to change default initialization parameters. These parameters are explained in the Transaction Subsystem Reference Manual. As mentioned previously,

when TRANEX1 completes initialization, the run time executive is loaded via the loader initiated from the OVERLAY macro in TRANEX1. Subroutines, FET, buffers, and common decks from TRANEX1 follow, in the order of their occurrence in the source.

RA+10B - Pointers (set by TRANEX1)

RA+101B INIT code

FETS – RECOVERY file, JOURNAL FILE Ø FET, and Data Manager Input/Output FET

SETL – Set table locations and lengths (attach JOURNAL FILE O)

IDM – Initialize Data Manager (DM)

ABJ – Allocate buffers for JOURNAL files (other than Ø)

LTL – Load task library directories

ATT – Attach POOL, TRACE and JOURNAL files

XXJ – Initialize journal files for data base XX.

ICRT – Initialize Copied Record Address Table (CRAT)

ANT – Attach NETWORK DESCRIPTOR file

DIE – Process DM Error Messages

Messages – DM Error Messages

SDT – Set Data Base Table

Initialization Parameters

FET for DBID file

Error Messages for SETL

FET for task library file

Common Decks including:

COMBINT – Initialize Data Manager

Circular Buffers

SETK – K-Display Initialization

K-Display Command Processors

K.SCP = N

K.CMB = N

K.CRS = Terminal Name

K.REC = AA

K.MFL

K. TLF = Task library file name

K. MDM = N

K. DB1 = AA

K-Display Subroutines

Common Decks

The character "V" as the first character of a symbolic name is used to indicate a table or buffer pointer. The values are initialized by TRANEX1 and used throughout TRANEX via the opdef* calls. The TRANEX preset code (PRE) actually performs the instruction modification for the table addresses. In other words, the variable length tables are assigned addresses during initialization (TRANEX1) and referred to thereafter with 30-bit increment instructions without the need of picking up a pointer (TRANEX). The names and locations of the pointers are listed in Table 14-1.

TABLE 14-1. TABLE AND BUFFER POINTERS

Word	Name	Meaning
10	VNSCP	Number of subcontrol points
11	VNCMB	Number of communication blocks
12	VSTS	FWA of terminal status table
13	VNTST	Number of entries in terminal status table
14	VMDM	Multiple for DM buffers
15	VLSP	Address of last subcontrol point
16	VATL	Address of Active Transaction List (ATL)
17	VFSCP	FWA of sub-CP allocatable storage
20	VCBRT	CB (Communication Block) storage allocation bit maps
21	VCBSA	Start of communication blocks
22	VTLD	Start of task library directories
23	VEDT	Base address of descriptor tables
24	VPOTT	Start of buffer area for Data Manager
25	VMFL	Maximum FL for subcontrol points, 40K≤ VMFL ≤ 300K (different than K. MFL)
26-31	VSDB	Data Base Names specified by operator (used instead of DBID) 50 → 240K.
32	VTFL	Task library file name
33	VREC	Recovery flag
34	VCRAT	FWA of Copied Record Address Table (CRAT)
35	VCRS	CRAS terminal name

* $TA_{(x)}$ opdef used in the following manner:
TA1 variable name ex: VSTS is expanded as SA1 VSTS
SA1 x 1 + proper offset to table. But not modified as two instructions but as one by PRE as: SA1 (VSTS) + proper offset to table.

The flowchart shown in Figure 14-5 outlines the routine INIT which performs the initialization for the executive and for the data manager. The flowchart shows an overview of the Transaction Executive initialization process. The tables and buffers are set up in subroutine SETL. Following the flowchart, Figure 14-6 provides an overview of TRANEX memory, showing the order of the tables and buffers established during initialization. Subsequent discussion explains each table and buffer.

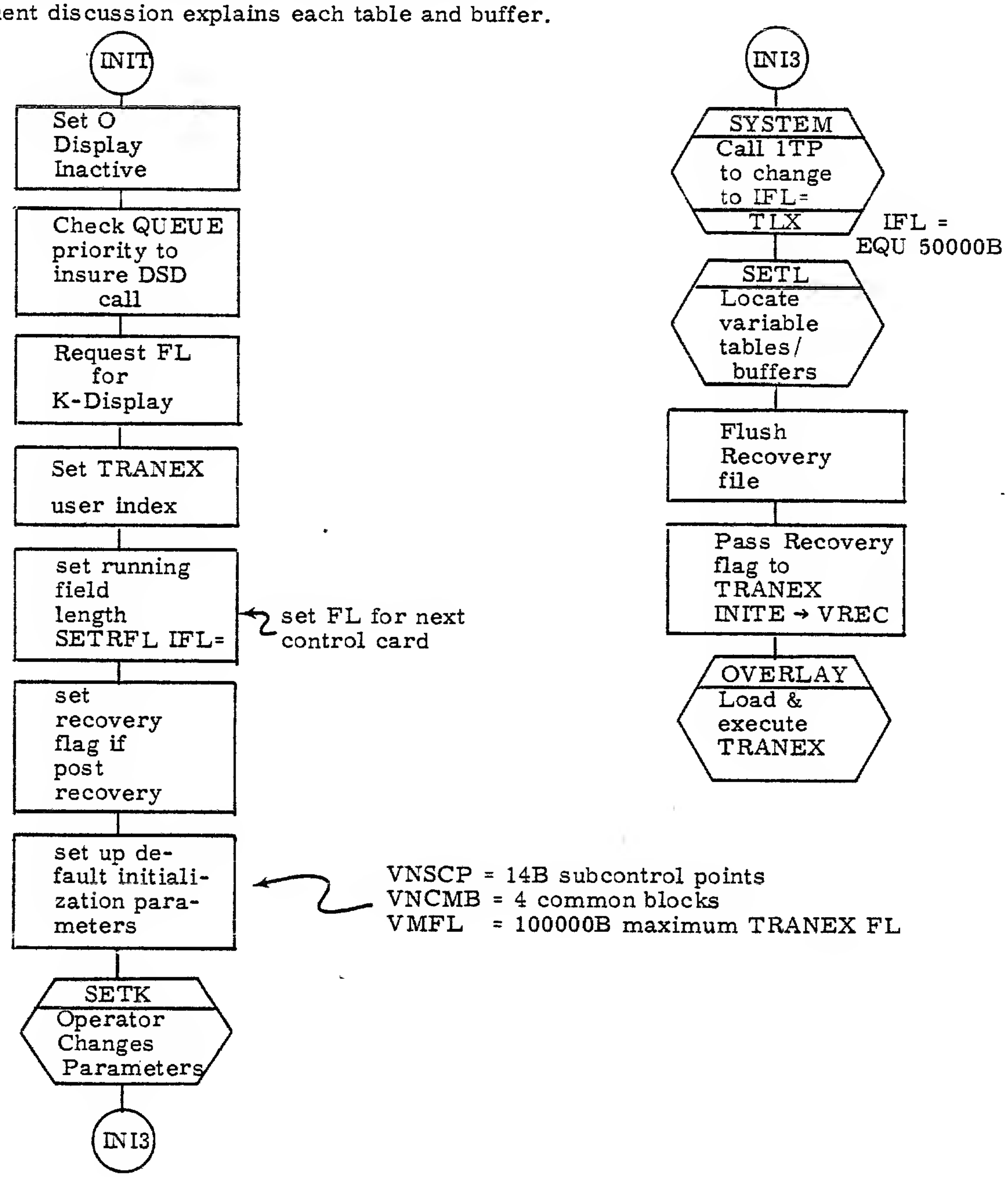


Figure 14-5. INIT Flowchart

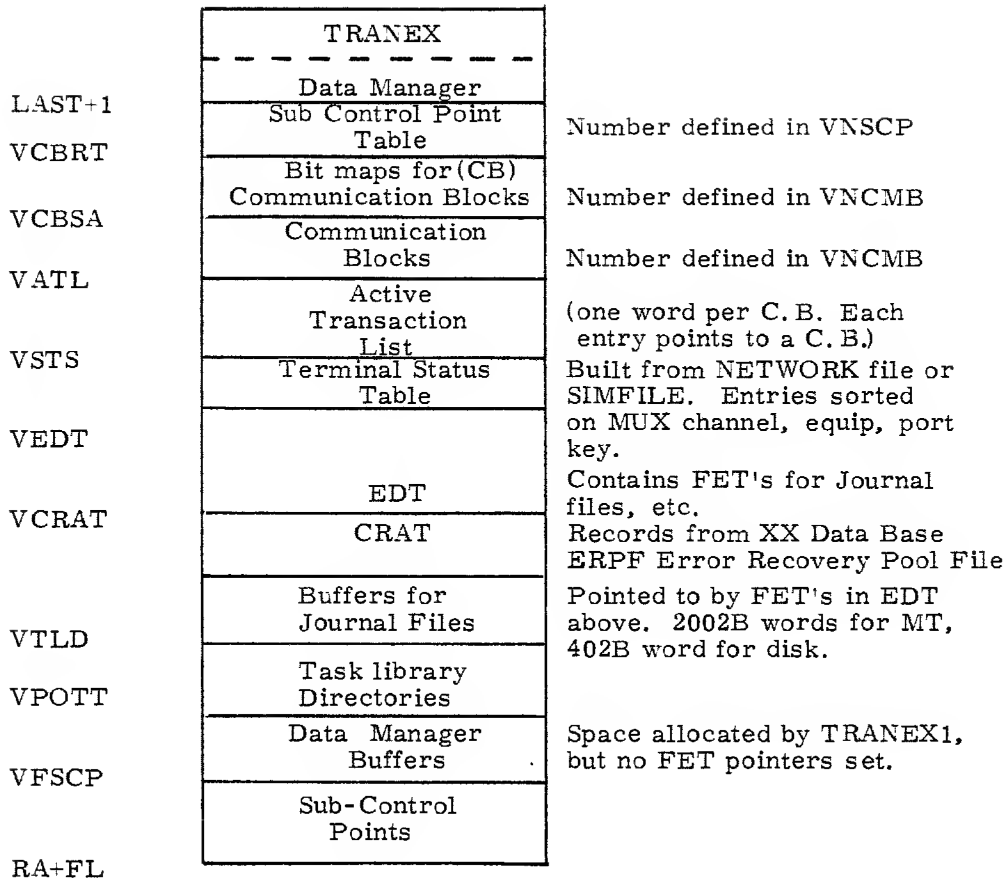


Figure 14-6. Buffers and Tables of TRANEX

14.4.1 Sub-Control Point Table

The structure of the sub-control point table as established by SETL is shown in Figure 14-7.

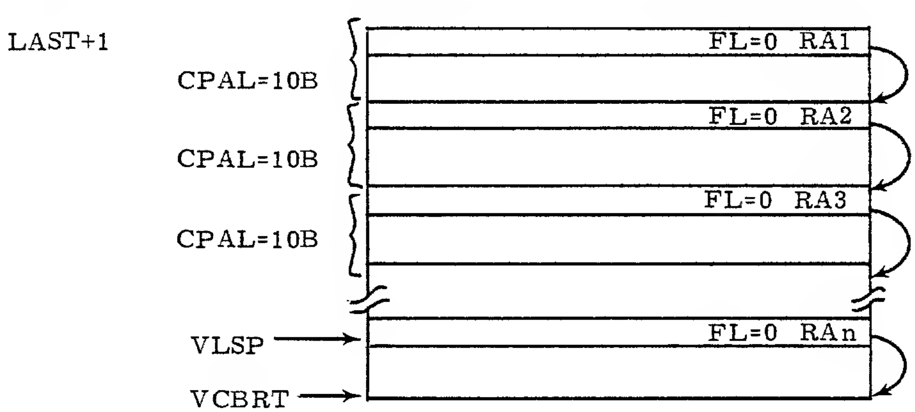


Figure 14-7. Sub-Control Point Table

RA is relative to TRANEX1's RA. The format of the sub-control point table entry is shown in Figure 14-8.

(Status Words)					
Word 1	flags	FC		FL	RA
Word 2	flags	NC		EP	CC
Word 3	NM		TS	LS	NS
Word N	flags				CBA

Figure 14-8. Sub-Control Point Table Entry Format

14.4.1.1 Status Word 1

Flags - bit	Meaning
59	if set, the S. C. cannot be moved
58	if set, this S. C. can be released if storage needed

FC — Available free core after sub-control point

FL — Sub CP field length

RA — Sub CP reference address (relative to TRANEX RA)

14.4.1.2 Status Word 2

Flags - bit	Meaning
59	flags a system task which gets entire comm. block
58	if set, task code is reusable
57	if set, task is CM resident
56	recall status bit
55	not used
54	if set, task is to be aborted

NC — Number of communication blocks at sub C. P.

EP — Entry point address

CC — Address of status word for C. B. now in execution

14.4.1.3 Status Word 3

NM — Task directory index

TS — Time slice limit

LS — Last sub-control point

NS — Next sub-control point

14.4.1.4 Status Words N

Flags - bit — Meaning

Flags - bit	Meaning
59	set if communication block is present at sub C. P.
54	set if initial comm. block

CBA — FWA of Communication Block

The length of the sub-control point entry is 10B words. Thus, there are 5 words of type N. The length of an entry is defined by CPAL and must be a multiple of 10B (i. e., 10B, 20B, 30B, etc.).

14.4.2 Communication Blocks

Communication blocks are set up by SETL merely by reserving CMBL* NCMB words. CMBL is the length of one entry and is equal to 69+5 = 74 (a 5-word system header and 69 words of data). NCMB is the number of communications blocks (4 by default). Although not written during initialization, the format of the 5-word system header is shown in Figure 14-9.

COMMUNICATIONS BLOCK SYSTEM HEADER

```
            5         4         3         2         1
       987654321098765432109876543210987654321098765432109876543210
      --------------------------------------------------------------
W1    /   CP     /DRMA    /  SEQ               /       DRC        /
      --------------------------------------------------------------
W2    /        TSO       /RS/US/     TST        /       CBA        /
      --------------------------------------------------------------
W3    /    1T     /     2T      /    3T     /    4T     /    5T    /
      --------------------------------------------------------------
W4    /ABC        /       LWA        /          /       FWA        /
      --------------------------------------------------------------
W5    /       QD                                / /OT /     QI     /
      --------------------------------------------------------------
```

WORD 1

CP	–	CPU PRIORITY
D	–	ALLOWED TO MAKE DBA REQUESTS
R	–	RECALL ON ALL OUTSTANDING D. M. REQUESTS
M	–	AT LEAST ONE MESSAGE WAS SENT TO TERMINAL
A	–	TRANSACTION CHAIN HAS ABORTED
SEQ	–	PRIMARY SEQUENCE COMMUNICATIONS BLOCK ADDRESS
DRC	–	DBA REQUESTS CURRENTLY OUTSTANDING

Figure 14-9. Communications Block System Header

```
WORD 2
    TSO  -  TERMINAL ORDINAL
    RS   -  TERMINAL DATA BASE READ SECURITY LEVEL
    WS   -  TERMINAL DATA BASE WRITE SECURITY LEVEL
    TST  -  ADDRESS IN TST FOR TERMINAL
    CBA  -  COMMUNICATIONS BLOCK ADDRESS

WORD 3
    1T   -  NEXT TASK SCHEDULE
    2T   -  2ND TASK IN CHAIN TO SCHEDULE
    3T   -  3RD TASK IN CHAIN TO SCHEDULE
    4T   -  4TH TASK IN CHAIN TO SCHEDULE
    5T   -  5TH TASK IN CHAIN TO SCHEDULE

WORD 4
    A    -  VALID DSDUMP REQUEST (A=1)
    B    -  DUMP EXCHANGE PACKAGE (B=1)
    C    -  DUMP DATA BASE BUFFERS (C=1)
    LWA  -  LAST WORD ADDRESS OF TASK DUMP
    FWA  -  FIRST WORD ADDRESS OF TASK DUMP

WORD 5
    QD   -  QUEUE DESIGNATOR (SEE K. DSDUMP)
    OT   -  ORIGIN TYPE VALUE OF QUEUE DESTINATION
    QI   -  QUEUE DESTINATION INDICATOR
```

Figure 14-9. Communications Block System Header (Continued)

14.4.3 Active Transaction List

The Active Transaction List (ATL) as established by SETL contains a 1-word entry for each communication block. Each ATL entry contains a pointer to a communication block. The format of the ATL entry is shown in Figure 14-10.

```
                  ATL  ACTIVE TRANSACTION LIST

         5         4         3         2         1
987654321098765432109876543210987654321098765432109876543210
------------------------------------------------------------
/   NT      /     PT     /    /          CBA                /
------------------------------------------------------------

 NT        NEXT TASK IN QUEUE CHAIN (BIASED BY +1)
 PT        PREVIOUS TASK IN QUEUE CHAIN (BIASED BY +1)
 CBA       ADDRESS OF COMMUNICATIONS BLOCK
```

Figure 14-10. Active Transaction List

14.4.4 Terminal Status Table

The Terminal Status Table (TST) contains a 2-word entry for each terminal described in the NETWORK file or SIMFILE (Diagnostic). The list of entries is sorted according to multiplexer channel, equipment, and port key. For a description of the NETWORK file, consult Part IV, Section 3 of the Installation Handbook. The format of the TST entry is shown in Figure 14-11.

```
                      TST   TERMINAL STATUS TABLE

              5         4         3         2         1
         987654321098765432109876543210987654321098765432109876543210
         ------------------------------------------------------------
   W1    / DO  CH /EQ/ PT  /RS/US/    DB       /    UA               /
         ------------------------------------------------------------
   W2    /        TN                                 /    NT         /
         ------------------------------------------------------------

          WORD 1
              D    -   TERMINAL DOWN
              O    -   TERMINAL ON/OFF
              CH   -   MULTIPLEXOR CHANNEL
              EQ   -   MULTIPLEXOR EQUIPMENT
              PT   -   MULTIPLEXOR PORT
              RS   -   DATA BASE READ SECURITY LEVEL
              US   -   DATA BASE UPDATE SECURITY LEVEL
              DB   -   DATA BASE TERMINAL IS VALIDATED TO USE
              UA   -   USER AREA

          WORD 2
              TN   -   TERMINAL NAME
              NT   -   NUMBER OF TRANSACTIONS RECEIVED FROM TERMINAL
```

Figure 14-11. Terminal Status Table

After SETL completes the initialization of the tables mentioned previously, routine IDM is initiated to initialize the data manager using the remaining field length. IDM attaches the data base identification file (DBID). The contents of the file are read into core (error if not enough core) and written to the RECOVERY file. The entries in the file contain data base names for the EDT files which must be attached. However, before proceeding, subroutine SDT is executed to allow the operator (via the K-display) to specify up to three data base names. Specifying certain names negates the use of DBID. The main loop for data manager initialization starts at location IDM3 and continues through IDM4 and IDM5. The procedure is outlined in the following steps:

1. Attach XXJ file for this data base. This file provides the user's account number and password which are stored in the EDT header.

2. Call subroutine INT in common deck COMBINT. This routine attaches the journal files described in the XXJ file attached in step 1. Trace and pool files (XXTFIL and XXERPF) are assigned FET's. The 5-word EDT header is initialized followed by the EDT entries for this data base as specified in file XX (XX=two character data base name). (The EDT header is described later.)

3. A call to subroutine ATT attaches the pool and trace files (XXERPF and XXTFIL).

4. Call XXJ again. XXJ will establish FET's for the journal files (maximum of 3), call ATT to attach them, and update EDT header word 2 and 3.

The preceding processing continues for all known data bases. Next, the Copied Record Address Table (CRAT) is initialized by subroutine ICRT. The subroutine reads the Error Recovery Pool Files (XXERPF) for the various data bases. Any records found in these files are placed into the CRAT. The CRAT is defined to be CRATL words long (currently CRATL=100B). If more records exist than what will fit into the 100B word table, TRANEX1 aborts.

After initializing the CRAT, IDM calls subroutine ABJ to allocate circular buffers for journal files. Tape and disk buffer sizes are defined by symbols TAPL and DSKL, respectively.

Currently, TAPL = 2002B, and DSKL = 402B.

Next, the last subroutine, LTL, is called. This subroutine loads the system task library directory, TASKLIB, and XXTASKL (XX=data base), the directory for each data base. These directories have been created by subsystem utility, LIBTASK, and occupy the last record of the task library. Reference the KRONOS 2.1 Reference Manual for the format of the directory.

Finally, IDM determines the amount of buffer space required based on the number of sub-control points, and sets the starting address for the sub-control points. Control returns to INIT5+1 which flushes the recovery file and loads the main program, TRANEX. The loader performs the loading of TRANEX and begins execution at the preset routine PRE.

Figures 14-13 and 14-14 contain tables set up by INT for data manager initialization.

14.4.5 EDT Structure

The EDT format is shown in Figure 14-12.

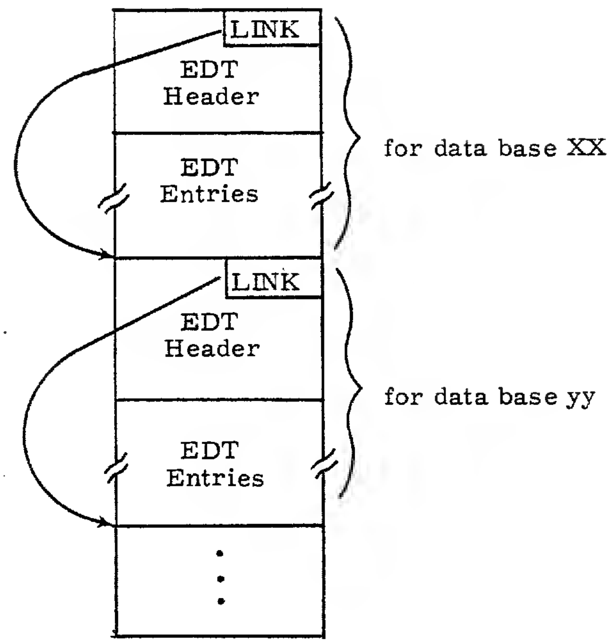


Figure 14-12. EDT Format

Figure 14-13 describes the format of the EDT header and the EDT entry. (The EDT entry is the EDT entry created by DATADEF).

```
              5         4         3         2         1
         987654321098765432109876543210987654321098765432109876543210
         ------------------------------------------------------------
VEDT1    / DB         /            /EDTCNT            / LINK        /
         ------------------------------------------------------------
VEDT2    /JORCN/                   /JORADP            /TRCADR       /
         ------------------------------------------------------------
VEDT3    /       USERNM                               /USINDX       /
         ------------------------------------------------------------
VEDT4    /       PASSWD                               /             /
         ------------------------------------------------------------
VEDT5    /            /TLDFWA          /      /TLDLWA         /
         ------------------------------------------------------

           WORD 1
              DB      - DATA BASE NAME
              EDTCNT  - NUMBER OF EDTS (PRESENT ONLY IN FIRST HEADER)
              LINK    - POINTER TO NEXT EDT
```

Figure 14-13. EDT Header and Entries

WORD 2
JORCN – NUMBER OF JOURNAL FILES (MAXIMUM OF 3 PER DB)
JORADR – ADDRESS OF FIRST JOURNAL FILE FET
TRCADR – ADDRESS OF TRACE FILE FET

WORD 3
USERNM – USER NUMBER (USED TO ATTACH MULTIPLE TLDS)
USINDX – USER INDEX (TO ATTACH POOL, JOURNAL, TRACE, DB, AND DATA BASE FILES)

WORD 4
PASSWD – PASSWORD

WORD 5
TLDFWA – FWA OF DBTASKL (NAME OF PARTICULAR TLD)
TLDLWA – LWA OF DBTASKL

EDT ELEMENT DESCRIPTOR TABLE

```
                 5         4         3         2         1
          987654321098765432109876543210987654321098765432109876543210
          ------------------------------------------------------------
DES1      / DB NAME  /          / EDT LENGTH    / NUMBER OF FILES /
          ------------------------------------------------------------
DES2      /      / ADDR GROUP LIST / ADDR SEARCH TABL/ ADDR FNT   /
          ------------------------------------------------------------
FNT1      / FILENAME                                / A M /  /TD TYPE/
          ------------------------------------------------------------
FNT2      / A M BIAS             /RS/US/LENGTH ELST/ ADDR ELIST     /
          ------------------------------------------------------------
FNT3      / FWA FET     / NUMBER OF RECORDS   / RECORD SIZE        /
          ------------------------------------------------------------
FNT4      / CHAIN KEY / PTRS ORD   /ORD PTR FNT/ FILE LOCK I. D.    /
          ------------------------------------------------------------
          /   EOI     /AP/LN     / IN          /P/LN      /  IN     /

ELE       / NAME      /TY/FM/ FB  /RS/US/  LENGTH   / WRD ORD       /
          ------------------------------------------------------------
          / NAME      /TY/FM/OPEN/RS/US /LENGTH   / L ORD           /
          ------------------------------------------------------------
          / NAME      /TY/FM/ FB   /RS/US/ LENGTH  / SER ORD        /
          ------------------------------------------------------------
GLT       / EL ORD  /ED ORD    /EL ORD      /EL ORD      / EL ORD  /
          ------------------------------------------------------------
STL1      / PIO      / DFO      / OPEN                / WRD ORD / K TY/
          ------------------------------------------------------------
STL2      / PKSO SDF / PKSO PDF / SDFOE    /          / WRD ORD / K TY/
          ------------------------------------------------------------
STL3      / SDFIO    / SDFO      / OPEN               / WRD ORD / K TY/
          ------------------------------------------------------------
```

Figure 14-13. EDT Header and Entries (Continued)

-DES1-	
DB NAME	DATA BASE NAME.
EDT LENGTH	LENGTH OF ELEMENT DESCRIPTOR TABLE IN CENTRAL MEMORY WORDS.
NUMBER OF FILES	NUMBER OF FILES IN THIS DATA BASE.
-DES2-	
ADDR GROUP LIST	RELATIVE ADDRESS OF GROUP LIST TABLE.
ADDR SEARCH TABL	RELATIVE ADDRESS OF SEARCH TABLE.
ADDR FNT	RELATIVE FWA OF FILE NAME TABLE.
-FNT1-	
A M	ACCESS METHOD ORDINAL.
T	TRACE BIT.
D	DUAL RECORD BIT.
TYPE	FILE TYPE.
-FNT2-	
A M BIAS	BIAS VALUE FOR ACCESS METHOD.
RS	READ SECURITY OF RECORDS IN FILE
US	UPDATE SECURITY OF RECORDS IN FILE
LENGTH ELST	LENGTH OF ELEMENT LIST IN CENTRAL MEMORY WORDS.
ADDR ELIST	RELATIVE ADDRESS OF ELEMENT LIST.
-FNT3-	
FWA FET	ABSOLUTE ADDRESS OF FET FOR THIS FILE.
NUMBER OF RECORDS	NUMBER OF RECORDS IN A PRE-ALLOCATED FILE.
RECORD SIZE	RECORD SIZE IN CENTRAL MEMORY WORDS.
-FNT4-	
CHAIN KEY	ORDINAL OF ELEMENT IN THE CHAINED FILE WHICH IS THE KEY TO THE OWNER FILE. THIS KEY IS THE COMMON LINK OF ALL RECORDS IN THE CHAIN.
PTRS ORD	ORDINAL OF ELEMENT IN THE OWNER FILE WHICH CONTAINS THE KEYS TO THE FIRST AND LAST RECORDS OF THE CHAIN.
ORD PTR FNT	RELATIVE ADDRESS OF THE FILE NAME TABLE ENTRY OF THE OWNER FILE.
-FNT5-	
EOI	CURRENT EOI OF FILE.
A	ADD/PURGE/RECHAIN IN PROGRESS.
P	FILE HAS AT LEAST ONE POOLED RECORD.
IN	INDEX FOR FILES *CRAT* ENTRIES.
LN	NUMBER OF POOLED RECORDS.
S	SECONDARY FILE (IF DUAL RECORDED) HAS AT LEAST ONE POOLED RECORD.
IN	INDEX FOR SECONDARY FILES *CRAT* ENTRIES.
LN	NUMBER OF POOLED RECORDS.

Figure 14-13. EDT Header and Entries (Continued)

-ELE-	
NAME	THREE CHARACTER ELEMENT NAME.
TY	TYPE OF ELEMENT.
FM	FORM OF ELEMENT.
FB	FIRST BIT (INDICATES POSITION WITHIN WORD).
RS	READ SECURITY CODE.
US	UPDATE SECURITY CODE.
LENGTH	LENGTH OF ELEMENT IN BITS.
WRD ORD	RELATIVE ADDRESS OF ELEMENT IN DATA FILE.
L ORD	RELATIVE ADDRESS OF GROUP LIST TABLE.
SER ORD	RELATIVE ADDRESS OF SEARCH TABLE.
-GLT-	
EL ORD	RELATIVE ADDRESS OF ELEMENT TABLE ENTRY FOR THIS SUB-ELEMENT OF THE GROUP.
-STL1-	
PIO	RELATIVE ADDRESS OF FNT ENTRY FOR THE PRIMARY INDEX FILE. THIS FIELD IS SET ONLY IF DFO IS AN INDEXED FILE.
DFO	RELATIVE ADDRESS OF FNT ENTRY FOR THE PRIMARY DATA FILE.
K TY	KEY TYPE.
-STL2-	
PKSO SDF	RELATIVE ADDRESS OF SEARCH TABLE ENTRY WHICH CONTAINS THE PRIMARY KEY INFO FOR THE SECONDARY DATA FILE.
PKSO PDF	RELATIVE ADDRESS OF SEARCH TABLE ENTRY WHICH CONTAINS THE PRIMARY KEY INFO FOR THE PRIMARY DATA FILE.
SDFOE	RELATIVE ADDRESS OF THE ELEMENT IN THE SECONDARY DATA FILE WHICH IS THE PRIMARY KEY TO THE PRIMARY DATA FILE.
-STL3-	
SDFIO	RELATIVE ADDRESS OF THE FNT ENTRY FOR THE PRIMARY INDEX TO THE SECONDARY DATA FILE. THIS FIELD IS SET ONLY IF SDFO IS AN INDEXED FILE.
SDFO	RELATIVE ADDRESS OF FNT ENTRY FOR THE SECONDARY DATA FILE.

Figure 14-13. EDT Header and Entries (Continued)

14.4.6 Task Library Directory

The task library directory header is shown in Figure 14-14.

```
           5              4              3              2              1
     987654321098765432109876543210987654321098765432109876543210
     ------------------------------------------------------------
VTLD-4  /                  /      TLDS          /      /      TLDL        /
     ------------------------------------------------------------
VTLD-3  / TLD              /      /      BOOT          /      NUMC        /
     ------------------------------------------------------------
VTLD-2  /          DATE                                                   /
     ------------------------------------------------------------
VTLD-1  /          NAME                                                   /
     ------------------------------------------------------------
```

VTLD-4

- TLDS – SIZE OF ALL TLDS COMBINED
- TLDL – LENGTH OF CURRENT TLD

VTLD-3

- TLD – 3 CHARACTERS USED TO VERIFY HEADER AS TLD TYPE
- BOOT – NUMBER OF ENTRIES IN DIRECTORY BOOT
- NUMC – NUMBER OF CORE RESIDENT TASKS

VTLD-2

- DATE – YY/MM/DD.

VTLD-1

- NAME – NAME OF TLD FILE.

Figure 14-14. Task Library Directory

14.5 TRANEX - Run Time Transaction Executive

TRANEX is loaded at RA+101B by the loader which begins execution at the preset routine, PRE. In general, PRE completes the initialization started by TRANEX1. This preset routine functions in the following sequence of steps.

1. Call subroutine SETA to modify the 30-bit increment instructions used throughout TRANEX binaries. This eliminates the need for reading up pointer words (V-words) when referencing tables.

2. Call PVV to set variable values, such as maximum field length (MFL), current field length (CURFL), and available central memory allocatable within TRANEX (AVAILCM).

3. Call PCR to set the CRAS terminal ordinal in CSMC for routines which send messages to the CRAS terminal.

4. Call LIT to load the initial task from system task library to sub-control point one. Initial task remains at sub-control point one as long as TRANEX runs.

5. Call LCT to read task library directories and load CM-resident tasks at sub-control points. If more tasks than sub-control points available, abort.

6. Call IJF to position each journal file to EOI and write a lable containing the current date.

7. Read date and time and real-time clock. Clear message line 2, and send version number to console.

8. Call SIC to initialize inter-control point transfers.

9. Jump to TMDC to begin main processing.

A memory map of TRANEX is showin in Figure 14-15. Notice that three SEG pseudo instructions are used in the assembly of TRANEX. Their purpose is to allow COMPASS to write partial binaries during assembly. Thus, less core is required by COMPASS to perform the assembly. This is done since the four blocks are quite large. The first block is about 4500 cards, the second is about 2900 cards, the third is about 7500 cards, and the fourth is about 1500 cards.

The symbol TRFL is defined at the end of subroutine TRI and is rounded up to the nearest 100B. The core from this point to the end (RA+FL is shown in Figure 14-6) is written to a rollout file by subroutine TRO when transaction activity stops.

14.5.1 Subroutine TRI will read the file, thus rolling the field length back into the TRANEX control point. This occurs when transaction input is received by subroutine PRIN or when the rollout time slice (TROTL) has elapsed (to ensure time-originated tasks are activated). Currently, TRFL=600B which is the TRANEX "idle" field length.

Location ENDT marks the end of the TRANEX run time code and the beginning of the fixed length buffers. Location LAST marks the end of the buffers and the beginning of the tables and buffers set up by TRANEX1. The fixed length buffers and their sizes are listed in Table 14-2.

TABLE 14-2. BUFFERS AND LENGTH

Buffer.	Length
JBUFO - Journal File	1201B
DIBF - D. M. Input FET	10B
DOBF - D. M. Output FET	30B
OBUF - Output Buffer	401B
SBUF - Scratch Buffer	100B

Time Dependent Routine Control consists of one routine named TMDC. TMDC calculates elapsed time for various subroutine calls. If the time limit for a particular routine has been exceeded, that routine is called. Subroutines called by TMDC include:

PRIN – Process transaction input
SCHD – Schedule tasks

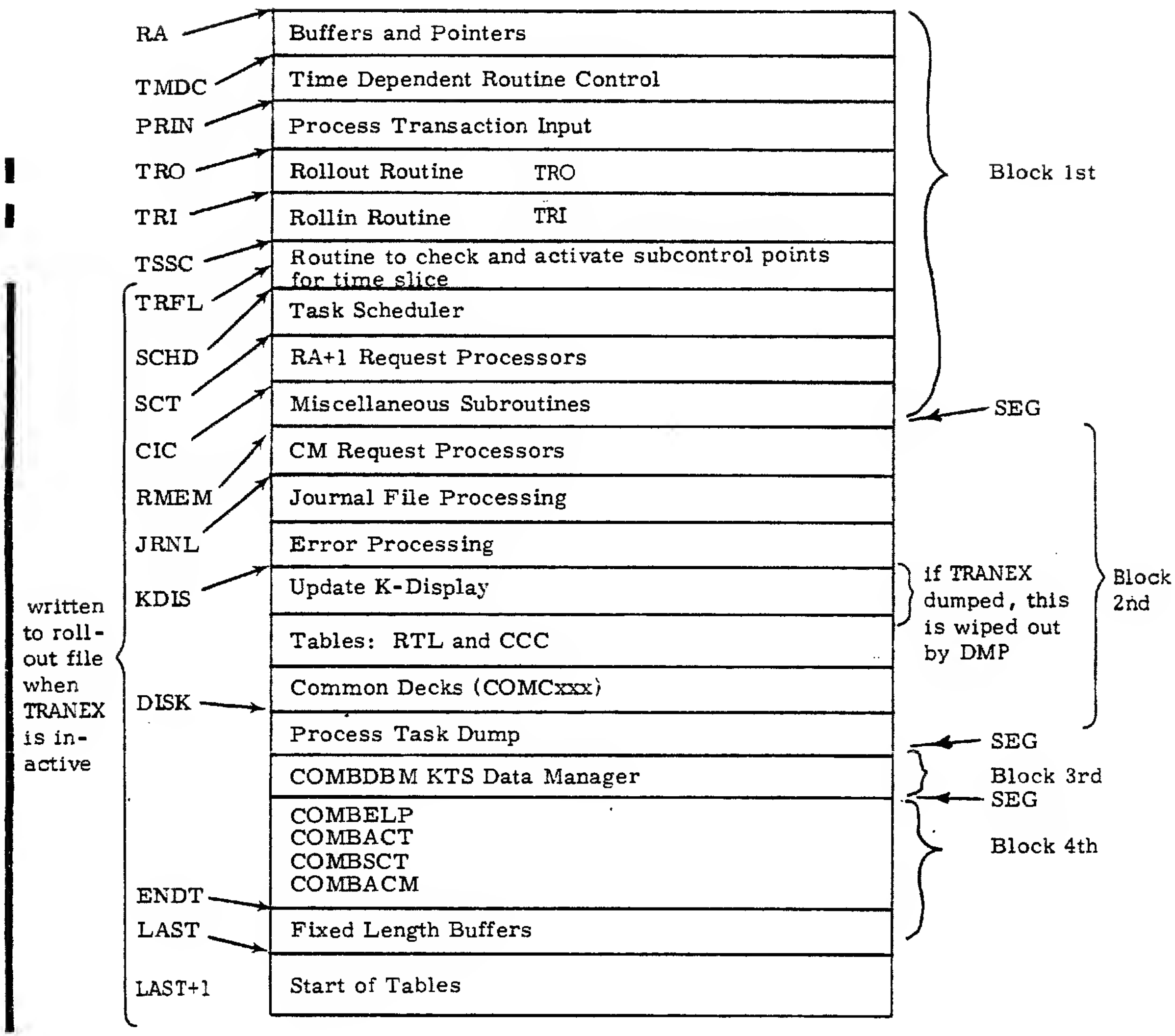


Figure 14-15. TRANEX Memory Map

DM requests are queued up and several issued at one time. These are the DIBF and DOBF queues which are like ITD queues, i.e. circular stacks. See figure 13-9

DCPT	–	Drop CPU for a task
KDIS	–	Update K-display
CORU	–	Check core usage
TRNA	–	Check transaction activity
JSTS	–	Write statistics to journal file
TSSC	–	Activate sub-control points

The Data Manager (DM) is called by subroutine TSSC only. The data manager returns control to TSSC at a location defined by symbol TSSC0. The batch data manager interface routine, BDMI, also adheres to this convention.

14.5.2 Again, referring to Figure 14-15, the RA+1 request processors include the following routines:

SCT	–	Schedule task
DBA	–	Data base access (build queue entry)
CTI	–	Call TRANEX interface
TIM	–	Request system time
MSG	–	Place message on line one

These routines process RA+1 requests from tasks executing at sub-control points. The general format of an RA+1 request is as follows:

59 — 42	41 — 36	35 — 0
NAME	R	ARGUMENTS
18	6	36

RA+1 request processing begins in subroutine TSSC at tag SRTN2. The RA+1 request is validated. NAME is equal to one of the five subroutine names mentioned above. R=20B if recall is desired. R is meaningful on DBA requests only, since all other requests are answered immediately by TRANEX.

14.5.3 The formats of the RA+1 requests for the particular routines are described subsequently.

14.5.3.1 SCT - Schedule a Task

59 — 42	41 — 36	35 — 18	17 — 0
SCT	0	FNC	ADDR

where:

FNC = Schedule function code (0-3)

ADDR = Parameter address (format not given)

FNC	Schedule Type
0	Task CEASE - end current task
1	NEWTRAN - start a new transaction
2	Call task with CEASE
3	Call task without CEASE - start an asynchronous task chain

14.5.3.2 DBA - Data Base Access

DBA (59)	R (41)	FNC (35–18)	ADDR (17–0)

where:

ADDR = FWA of data manager parameter area

FNC = Data manager function code

0-177: are handled by D. M. without special processing.

200+: are handled with recall.

14.5.3.3 CTI - Call TRANEX Interface

CTI (59)	R (41)	FNC (35–18)	ADDR (17–0)

where:

ADDR = FWA of parameter list

FNC = Function code (0-10)

0 = Send message to transaction terminal

1 = Make a journal file entry

2 = Check for a specific task chain still active

3 = Process terminal argument operation

4 = CMDUMP request

5 = DSDUMP request

6 = Return terminal status

7 = CRAS terminal K-display command

10 = Use task data field for K-display

Following is the format of the parameter list specified by ADDR for the functions of CTI.

<u>FNC = 0</u> Send Terminal Output

	59–48	47–30	29–18	17–0
ADDR	flags	MSG		NUM
ADDR+1	Terminal Name			

where:

flags – <u>bit</u> <u>meaning</u>

59 If set, send message to terminal specified in ADDR+1 else send to originating terminal.

58 CEASE task after sending message.

MSG – FWA of message

NUM – number of words in message (1-100B)

<u>FNC -1</u> Task Journal Request

	59–54	53(52)–36	35–18	17–0
ADDR		JN	NUM	MSG
	6	18	18	18

where:

MSG – FWA of block to be journaled

NUM – Number of words to write to journal file (max. = 2500D)

JN – Journal file number

<u>FNC -2</u> Check for Task Chain in System

59–42	41–18	17–0
	SEQ	STAT
18	24	18

where:

SEQ – Sequence number of transaction

STAT – Address of reply word. Reply word is set to zero if transaction is <u>not</u> in system.

The sequence number (SEQ) would be the number returned by TRANEX when the task issued a call to SCT function 1 (NEWTRAN).

FNC -3 Terminal Argument Operation

ADDR	Terminal Name		Return Address
ADDR+1	Value	Mask	

where:

Terminal name – terminal to be operated upon. If zero, originating terminal is assumed.

Return Address – Location in which to place result of operation (in addition to terminal table). Zero if no return desired.

Value – A value to be used to alter terminal arguments.

Mask – A 24-bit mask.

The USER ARGUMENT area (24 bits in each terminal table entry) is operated upon as follows:

RETURN=USER ARG=(USER ARG. AND.MASK).XOR.VALUE

Non-system tasks may only alter terminal arguments for those terminals that share the originating terminal data base.

FNC 4 CMDUMP

```
                  5         4         3         2         1
           987654321098765432109876543210987654321098765432109876543210
           --++-------------------------------------------------------
ADDR       /EDAB      /    LWA        /          /     FWA          /
           -++++------------------------------------------------------
ADDR+1     /      QD                             /      OT          /
           -----------------------------------------------------------
ADDR+2     /     AD       /                      / NF               /
           -----------------------------------------------------------
ADDR+N     /    FN                               /                  /
           -----------------------------------------------------------
```

Figure 14-16. CMDUMP

where:

- E – Dump exchange package
- D – Dump data manager buffers
- A – Use default exchange package parameter
- B – Use default data manager parameter
- LWA – Last word address of task to dump
- FWA – First word address of task to dump
- OT – Output Queue
- QD – Queue Destination
- AD – Address user called from
- NF – Number of specified files
- FN – Specified file name

FNC 5 DSDUMP

Default dump parameters setup for CMDUMP.

```
                5         4         3         2         1
        987654321098765432109876543210987654321098765432109876543210
        -++++-------------------------------------------------------
ADDR    /EDAB      /    LWA       /            /     FWA          /
        -++++-------------------------------------------------------
ADDR+1  /     QD                               /     OT           /
        ------------------------------------------------------------
```

Figure 14-17. DSDUMP

where:

- E – Dump exchange package
- D – Dump data manager buffers
- A – Use default exchange package parameter
- B – Use default data manager parameter
- LWA – Last word address of task to dump
- FWA – First word address of task to dump
- OT – Output Queue
- QD – Queue destination

FNC 6 Return Terminal Status

```
               5              4              3              2              1
98765432109876543210987654321098765432109876543210987654321O
------------------------------------------------------------
ADDR     /      / CODE/ LIST             /  LENG     /  RETURN         /
------------------------------------------------------------
ADDR+1   /  MASK                                                       /
------------------------------------------------------------
ADDR+2   /  CRIT                                                       /
------------------------------------------------------------
```

Figure 14-18. Return Terminal Status

where:

CODE = 0 If data base name field is to be searched.
= 1 If user argument field is to be searched.
= 2 If communication line field is to be searched.
= 3 If terminal name field is to be searched.

CRIT – Criterion value for search.

LENG – Number of words that list can hold.

LIST – FWA of list of returned terminal entries. If zero, no list is returned, but the number of found entries will be returned as specified below.

MASK – A value taken as a binary mask.

The field specified by CODE is examined in each terminal table entry by taking the logical product of the field and MASK, and then taking the logical difference of this product and CRIT. If this result is zero, the terminal entry is placed into LIST and the number of found entries is incremented.

FNC 7 CRAS Terminal K-Display Command

	59
ADDR	Start of Command

Valid commands are:

ASSIGN	OFFLINE
CHNGLIN	OFFTASK

DROP	OFFTERM
DSDUMP	ONLINE
DUMP	ONTASK
IDLE	ONTERM
JEND	RESPT
MAXFL	SWITCH
MESSAGE	

The command may occupy several words and must end with 12 bits of zero, just as a control card would.

FNC 10 Set K-Display to Run From a Task

	59 ... 0
ADDR	K-Display Control Word

This function is not performed if the K-Display is already being used by a task. This function simply replaces the current K-Display control word with the control word given. The format of the control word is as follows:

59 ... 36	35 ... 18	17 ... 0
KBUF	LS	RS
24	18	18

where:

KBUF – Address of keyboard buffer (8 words)

LS – Address of left screen control word

RS – Address of right screen control word

NOTE

Further information on K-Display usage is available in Section 19.

14.5.3.4 TIM – Request System Time/Date

59 ... 42	41 ...	35 ... 24	... 18	17 ... 0
TIM		OP		ADDR
18	6	12	6	18

where:

OP – Option (0-6)

ADDR – Address for response

14.5.3.5 MSG – Place Message on Line One

59	41	35	23	17 0
MSG	0	FNC	0	ADDR
18	6	12	6	18

where:

ADDR – Address of message to be displayed.

FNC – Function code. Currently only FNC=1 is supported.

Since the MESSAGE macro is used to display the message, the message must be in "C" (COMPASS) format.

Most of the RA+1 request processing routines described previously enter the TSSC subroutine upon completion of the request. TSSC is the subcontrol point supervisor which activates a subcontrol point via the XCHNGE macro. TSSC determines which subcontrol points are requesting the CPU and determines what servicing to schedule upon return of the CPU from a task (at SRTN). If there are any outstanding data manager requests, TSSC branches to DMGR (the data manager) before activating a subcontrol point. TSSC also monitors PP completion statuses and reinitiates routines when their PP call is complete. For example, TSSC restarts task loading after a PP has performed the load, or TSSC restarts non-buffered journal file processing as PP completion is sensed. Finally, at absolute time intervals, the system monitor drops the CPU from a subcontrol point so that control can be returned to the main loop for time-dependent processing. This time interval is defined by symbol TSL in TRANEX and is currently 200 milliseconds. Control returns at SRTN which checks error exit flags and RA+1 requests from the subcontrol point program. As mentioned earlier, if an RA+1 request is present, one of the processors described previously is executed.

TMDC calls the task scheduler (SCHD) every SCHTL milliseconds (currently SCHTL=60). SCHD searches the requested task list (RTL) for the highest priority task, requests enough core to run the task (via subroutine RMEM), and (if core is available) initiates loading of the task. The RTL is one of two tables assembled within TRANEX and not set up dynamically by TRANEX1 (reference Figure 14-15). The other table is a task load request stack with the name CCC. The RTL consists of 2-word entries and is currently 120B words long; while CCC consists of three 2-word entries with a zero-word terminator. The format of these two internal tables is shown in Figure 14-20.

```
         OP                      Response
          0       ACCUMULATED CPU TIME
                  5           4           3           2           1
        9876543210987654321098765432109876543210987654321098765432 10
        ------------------------------------------------------------
ADDR    /2 /  SECONDS                                 /MILLISEC. /
        ------------------------------------------------------------
          1       DATE
                  5           4           3           2           1
        987654321098765432109876543210987654321098765432109876543210
        ------------------------------------------------------------
ADDR    /  * YY/MM/DD. *                                           /
        ------------------------------------------------------------
          2       CLOCK
                  5           4           3           2           1
        987654321098765432109876543210987654321098765432109876543210
        ------------------------------------------------------------
ADDR    /  * HH.MM.SS. *                                           /
        ------------------------------------------------------------
          3       JULIAN DATE
                  5           4           3           2           1
        987654321098765432109876543210987654321098765432109876543210
        ------------------------------------------------------------
ADDR    /  0                         /  * YYDDD*                   /
        ------------------------------------------------------------
          4       SCOPE FORMAT REAL TIME (NOT SUPPORTED)
                  5           4           3           2           1
        987654321098765432109876543210987654321098765432109876543210
        ------------------------------------------------------------
ADDR    /2 /  SECONDS                                 / MILLISEC. /
        ------------------------------------------------------------
          5       REAL TIME
                  5           4           3           2           1
        987654321098765432109876543210987654321098765432109876543210
        ------------------------------------------------------------
ADDR    /  SECONDS               /  MILLISECONDS                   /
        ------------------------------------------------------------
```

Figure 14-19. OP Response

```
      OP                          Response

      6        PACKED DATE/TIME
                 5           4           3           2           1
     9876543210987654321098765432109876543210987654321098765432109876543210
    ---------------------------------------------------------------------
ADDR  / 0                    / Y-70/  MM / DD / HH / MM / SS     /
    ---------------------------------------------------------------------
```

Figure 14-19. OP Response (Continued)

RTL— Requested Task List

	59	47	41	29	23	17 — 0
W1	Name	FL		CP	MP	QL
W2	CRA		FRA		flags	

where:

Name	-	Task directory index
FL	-	Field length
CP	-	Current priority
MP	-	Maximum priority (future use)
QL	-	Queue length limit
CRA	-	Current ATL entry
FRA	-	First ATL entry
flags	-	bit / meaning

bit	meaning
2	ECS resident (future use)
3	CM resident
4	non destructive code
5	system task

CCC — Task Load Request Stack

	59	41	35	29	17 — 0
W1	CPA		TFL		SCP
W2	TLN			RDA	

where:

CPA	-	Subcontrol point area	TLN	-	Address of task library name
TFL	-	Task field length	RDA	-	Random disk address of task
SCP	-	Start of subcontrol point FL			

Figure 14-20. RTL and CCC Internal Tables

14.6 TRANEX2 – RECOVERY/END PROCESSING

In general, TRANEX2 performs the following operations:

- Flush buffered journal files
- Issue statistics to the dayfile
- Dump TRANEX field length
- Restart the subsystem

TRANEX2 is loaded over the K-Display processing code within TRANEX and currently cannot be expanded by more than 64B words.

Figure 14-21 is a flowchart of REC, the main control portion of TRANEX2.

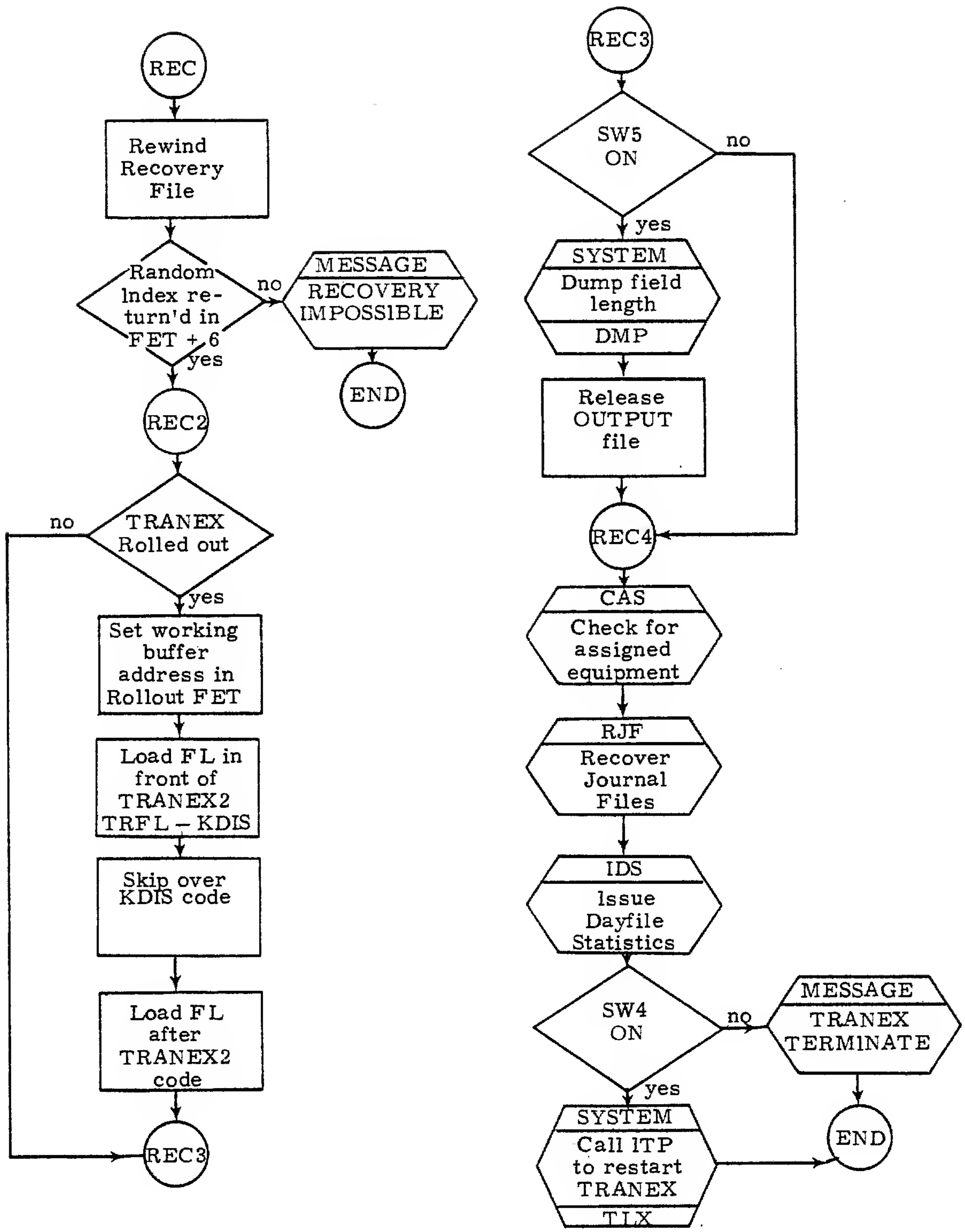


Figure 14-21. REC – Recovery Processor from TRANEX2

STIMULATOR 15

15.0 INTRODUCTION

The Stimulator is a program that acts as a load simulator on the KRONOS system. A load simulator is a procedure in which a small-scale computer (simulating communication network terminals) is programmed to transmit data into a large-scale computer using the required number of communication lines

The small-scale computer is programmed to accept input data and control traffic over the communication lines. The large-scale computer functions in the same manner as if the remote terminals were actually being used.

The organization of the CDC CYBER 70 computer system allows one PP program to communicate with another composed of a set of programs, one of which is a dedicated PP program transmitting data to and receiving data from the KRONOS communication driver over a data channel

The PP program actually simulates a 6676 or 6671 multiplexer That is, it responds to all function codes issued by the system communications driver in exactly the same manner as the multiplexer. To the communications driver and the KRONOS operating system, the PP program looks like a communication network of time-sharing users.

The Stimulator can be used to:

- Load the system for system checkout
- Gauge system performance
- Measure response times under varying loads

15.1 TELEX/TRANEX

The TELEX and TRANEX stimulators work in the same basic way Two PP's can communicate to each other via a channel; 1TD can transfer data to TELEX in such a way that TELEX can not tell if the input is from a live MUX or a simulated MUX.

Hence, stimulator receives standard session file information from a user-supplied input and multiplies them as directed by K display commands and transmits the session file across a channel to 1TD 1TD needs to realize the input is from a PP rather than a MUX only in the way he functions, reads, and writes across the channel. From that point on, analyzing and transmitting the data to TELEX is identical to a live MUX. Hence, most

of the driver and all of TELEX and TRANEX cannot discriminate real traffic from stimulator traffic.

The stimulators are composed of a CP program and a PP program The CP program controls the speed and repetitions of the session file. The PP program actually communicates with 1TD across a channel.

Normally 1TD just functions and inputs across the MUX channel

However, for stimulators, 1TD cannot input until the stimulator is ready to output and vice versa. Hence, 1TD and the stimulator PP are synchronized since 1TD covers all the MUX's in a prescribed sequence, and the PP doing the output will wait until the other PP performs an input These are sequenced so that 1TD will not get hung on input and miss some other MUX or stimulator input.

For example, in a very simplified way Figure 15-1 shows 1TD communication with a real MUX

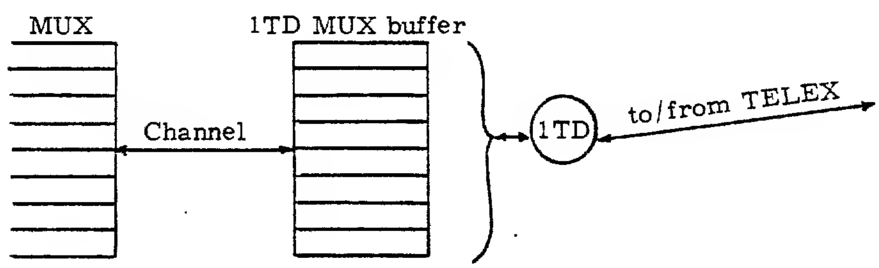


Figure 15-1. 1TD Communicating With MUX.

1TD inputs/outputs data from the MUX into its buffer at any time it desires and then processes the buffer.

Figure 15-2 shows 1TD communication with a stimulator PP.

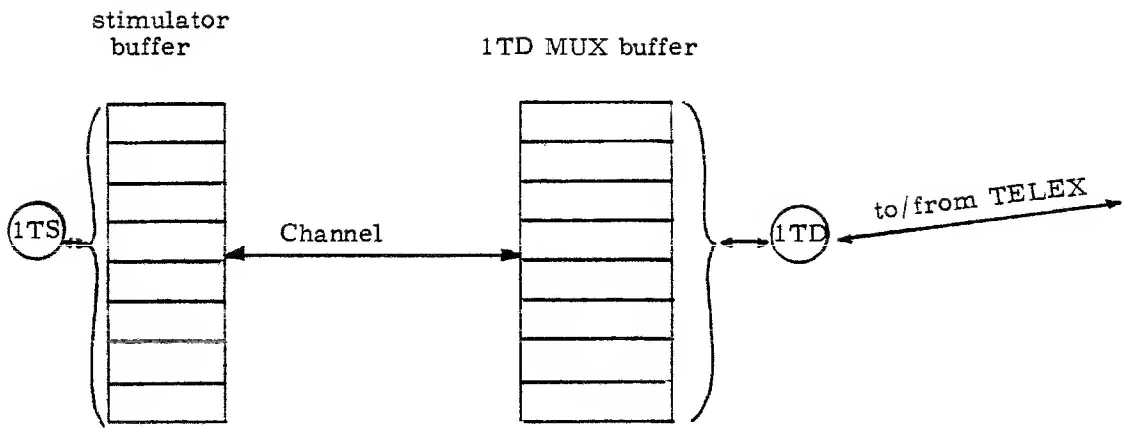


Figure 15-2. 1TD Communicating With PP.

1TD must input/output data from the stimulator only when the stimulator PP is correspondingly output/input across the channel. Once the input/output is complete, the buffer is handled in exactly the same manner as a true MUX buffer.

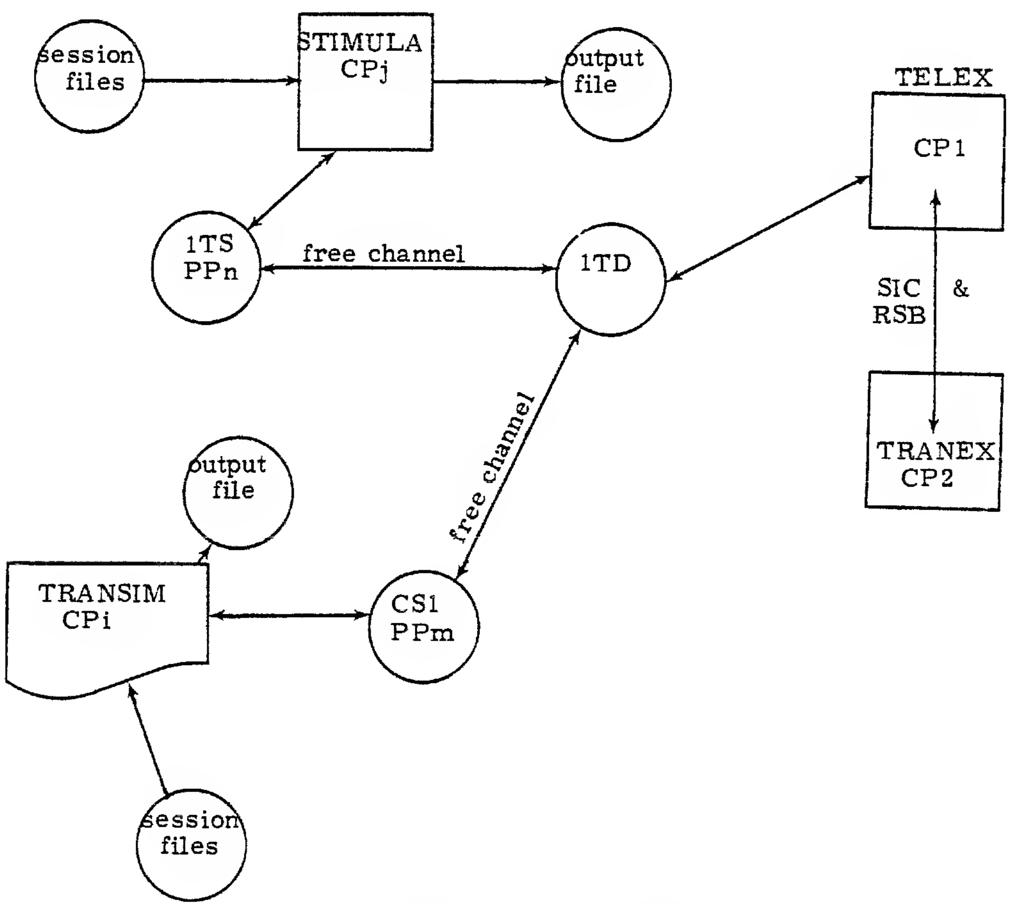


Figure 15-3. Stimulators

1. Referring to Figure 15-3, a session file is produced by the user. (In the case of TRANEX, a previously defined data base and task library must exist.

2. The file is read by the CP portion and is duplicated and time-controlled to send transactions or commands.

3. The PP portion receives the data from the CP portion, connects the free channel and transmits the data across the channel. This data looks identical to real MUX input.

4. 1TD recognizes this as a stimulator channel (by the MUX table) and inputs the data It then treats it as any normal MUX input and the rest of 1TD is normal.

5. TELEX receives the POTS, processing them, which can generate output POTS. Live and stimulated traffic are processed exactly the same.

6. 1TD picks them up in a normal way However, 1TD realizes the output is meant for a stimulator and outputs on the stimulator channel.

7. The PP portion collects the output and sends it to the CP portion.

8. The CP portion times responses to the stimulated traffic and collects other information for use by the analyst

The user can vary the extent of the load, and specify the manner in which the simulated load is to be run, together with the frequency and number of terminals logging in.

A test program to be run under the Stimulator must resemble a typical session at a time-sharing terminal or transaction terminal. Each test program includes:

- Terminal log-in
- A source file
- Any necessary terminal commands
- All necessary data
- Terminal log-off

The system processes the program the same as a program from a time-sharing terminal.

15.2 TELEX STIMULATOR

The TELEX Stimulator simulates up to 8 multiplexers and 512 TTY users. The Stimulator is quite useful in benchmarks and demonstrations, as a device for loading the system during Q.A., and for actually QA'ing TELEX after making enhancements

The Stimulator operates by actually feeding data into the TELEX POTs which are handled as though it is data from a TTY user. The Stimulator feeds data to TELEX at about the same speed as an actual user. Jobs are submitted to control points and the output is returned to the Stimulator.

15 2.1 Stimulator Requirements

Three items are necessary to allow the Stimulator to run:

1. Entry in the VALIDUX file.
2. A dummy 6676 entry in CMR = EST entry for a free channel must be specified.
3. A Stimulator input file on a permanent file.

15.2.2 Operation of the TELEX Stimulator

1. At deadstart time a dummy entry for a 6676 multiplexer must be made to CMR A sample entry would look like the following:

   EQ30=TT, OFF, 0, 1, CH, 0, 20. Refer to Part II, Section 4 of the Installation Handbook.)

   The CH = a free channel. This is very important. If there is any equipment on this channel, it will cause the Stimulator and TELEX to hang their associated PPs. After this entry is completed, KRONOS is ready to deadstart.

   The IPRDECK should also be checked to verify that AUTO is not selected. If it is, enter AUTO again to negate the selection.

2. Do a MODVAL run entering in legal account numbers and passwords that match those on the Stimulator input file. These entries must be made and made correctly or the Stimulator will not run. Figure 15-4, example 1 Stimulator input file requires the following entries in the VALIDUX file:

   a. Account Number = SKUJINS PW=JURIS

   b. Account Number = KIRKLND PW=CLAY

   Figures 15-5 and 15-6 Example 2 requires the following entries in the VALIDUX file:

   a. Account Number = BOBSIM1 PW=7744526

   b Account Number = BOBSIM2 PW=7744526

   c. Account Number = BOBSIM PW=7744526

3. Make the following entry, "UNLOCK". This unlocks core for the following entries to core if needed.

4. Enter EB. This gives you the console E display on the left hand CRT The E display is a display of the Equipment Status Table (EST). In the upper right hand corner of the display is the EST FWA. This address is needed to change the EST ordinal for the dummy entry of the 6676 multiplexer.

5. Enter "C4, EST FWA + ORDINAL NUMBER OF MULTIPLEXER". Take for instance that the EST FWA is 4600 and the ordinal entry of the 6676 in CMR is EQ30. The entry would be then C4, 4630. After this entry, the location 4630 is then displayed on the left hand scope.

6. It is now necessary to ensure that the EST entry is correct in central memory resident and if not, correct it.

   The detail format for the MUX is shown in Part II, Section 4 of the Installation Handbook. A general format is reproduced here.

EQord = TT, status, controller, 0, channel, 0, lines, for a real MUX

EQord = TT, status, controller, no, channel, 0, lines for a dummy MUX.

If the following entries were used in the CMR deck:

EQ30 = TT, OFF, 7, 0, 4, 0, 20

EQ31 = TT, OFF, 0, 1, 11, 0, 20

then the EST entry can be seen in the dump in Section 27 of this manual address 6630 and 6631.

6630 = 2000 0004 0020 6424 7000

6631 = 2001 0011 0020 6424 0001

The general format for the EST entry is:

6	6	6	6	12	1	11	3	3	6
20	CP number	0	channel no	Number of lines(ports)	* 2	TT	eq no	0	stim. type

20 = Device available for use.

CP number = CP currently assigned. (Normally TELEX will have this device assigned and CP number =1)

*2 Bit 23 = 0 = device status ON
1 = device status OFF

eq. no. = Controller number

stim type = 0 Real MUX
1 Time-sharing stimulator for either 6671 or 6676
2 TRANEX stimulator for 6676
4 TRANEX stimulator for 6671

Now, either entry can be dynamically created by entering data into core. The number of lines can be dynamically changed by entering data into core as follows.

6631, 2, 30. would make the Stimulator stimulate 30 lines.

The command format is*:

address, Byte number, 12-bit value, or
address, 60-bit value.

When the entries have been created to the user's satisfaction, he can continue.

7. Enter ONXX. CR with XX having the value of the EST ordinal number of the 6676 dummy entry. In the preceding examples EQ30 was used as the EST ordinal. Therefore, you would enter "ON30. CR. EQ31 was the stimulator EST ordinal so also enter ON31 CR.

* Refer to Section 3, Memory Entry commands of the Operator's Guide

```
BNCHMARK01v
SKUJINSv
JURISv
FORv
NEW, RALFv
10 PROGRAM RALF(INPUT, OUTPUT, TAPES=INPUT, TAPE6=OUTPUT)v
20 READ(5. 9) ANGLE1, ANGLE2, ANGLE3, ANGLE4v
30 RAD=57.29573v
40 A=ANGLE1/RADv
50 B=ANGLE2/RADv
60 C=ANGLE3/RADv
70 D=ANGLE4/RADv
80 S1=SIN(A)v
90 C1=COS(A)v
100 T1=TAN(A)v
120 CO1=1T1v
130 WRITE(6.1) ANGLE1, S1, C1, T1, CO1v
220 S2=SIN(B)v
240 C2=COS(B)v
250 T2=TAN(B)v
260 CO2=1/T2v
270 WRITE (6.1) ANGLE2, S2, C2, T2, CO2v
300 S3=SIN(C)v
320 C3=COS(C)v
330 T3=TAN(C)v
340 CO3=1/T3v
360 WRITE (6.1) ANGLE3, S3, C3, T3, CO3v
410 S4=SIN(D)v
430 C4=COS (D)v
450 T4=TAN(D)v
460 CO4=1/T4v
470 1FORMAT(*ANGLE=*, F4.0, *SIN=*, F6.2, *COS=*, F6.2, *TAN=*, F6.2v
480 +* COTAN=*F6.2)v
500 WRITE (6.1) ANGLE4, S4, C4, T4, CO4v
510 9FORMAT(4F4.0)v
520 CALL EXITv
530 ENDv
RUNv
30. 45. 60. 90.v
BYEv
    (this punch is not necessary for KRONOS V2.1)
0000000000000000000000000  (eor punch 7/8/9)
BNCHMARK02v
KIRKLNDv
CLAYv
BASv
NEWv
ROBINv
10 REM PROGRAM ROBINv
12 N1=9v
14 LET L1=1000v
16 LET K=Sv
18 LET N=1v
20 K=K+1v
22 IF K<=N1 THEN 30v
25 S10Pv
30 N=K+Nv
32 IF N>=L1 THEN 18v
34 L=0v
38 J=Nv
```

Figure 15-4 Stimulator Input Files, Example 1

```
40 J=J-1v
44 M=INT (J/K)v
48 B=J-K* Mv
50 IF B<>0  THEN 30v
55 LET J=J-Mv
58 L=L+1v
60 M=L-Kv
65 IF M<=0  THEN 40v
70 PRINT K,Nv
75 GO TO 18v
80 ENDv
RUNv
BYEv
|--------(this punch is not necessary for KRONOS V2.1)
0000000000000000000000000 (eor)
0000000000000000000000    (eof punch 6/7/8/9)
```

Figure 15-4. Stimulator Input Files, Example 1 (continued)

Example 2 needs the permanent files created by the following two decks.

```
STIMA,CM12000,T7777,P77.
ACCOUNT,BOBSIM2,7744526.
COPYBR(INPUT,TESTA)
SAVE,TESTA.
(eor punch=7/8/9)
00010 TEST A
00120 GHI
00130 KKK
00140 AAA
00150 BBB
00160 CCC
00170 DDD
00180 EEE
00190 FFF
00200 GGG
00210 HHH
00220 III
00230 JJJ
00240 LLL
00250 KKK
00260 MMM
00270 NNN
00280 OOO
00290 PPP
00300 QQQ
00310 RRR
00320 SSS
00330 TTT
00340 UUU
00350 VVV
00360 WWW
00370 XXX
00380 YYY
00390 ZZZ
910 NOMORE PLEASE
920 YOU NEED MORE
930 MY FRIEND
940 SOLD TO MLO
950 STICKS AND STONES
960 HURT LIKE HECK
970 AND SO DO HOLDERS
980 BUT WHAT CARE I SAY THIS MACHINE
990 TISS TOWARD THE END
1000  THE END PART 2 TEST  A
(eof punch=6/7/8/9)
STIM,CM12000,T7777,P77.
ACCOUNT,BOBSIM1,7744526.
COPYBR(INPUT,TESTB)
SAVE,TESTB.
(eor punch=7/8/9)
010 TESTB
00100 ABC
00110 DEF
00120 GJO
00130 KKK
00140 AAA
00150 BBB
00160 CCC
```

Figure 15-5. Stimulator Input Files, Example 2 Permanent Files Decks

```
00170 DDD
00180 EEE
00190 FFF
00200 GGG
00210 HHH
00220 III
00230 JJJ
00240 LLL
00250 KKK
00260 MMM
00270 NNN
00280 OOO
00290 PPP
00300 QQQ
00310 RRR
00320 SSS
00330 TTT
00340 UUU
00350 VVV
00360 WWW
00370 XXX
00380 YYY
00390 ZZZ
900 HI GANG
910 NANN
930 FRAN
950 FLILL
960 HOBOY
970 OUCH
980 980
990 NINE NINTY
1000 THE END
(eof punch=6/7/8/9)
```

Figure 15-5. Stimulator Input Files, Example 2
Permanent Files Decks (continued)

Example 2 Stimulator input file is created by the following deck on permanent file STIMFL.

```
STIMM,CM12000,T7777,P77,TP1.
ACCOUNT,BOBSIM,7744526.
COPYBR(INPUT,STIMFL)
SAVE,STIMFL
(eor punch)
ANSWERBACv
BOBSIM1v
7744526v
BASICv
OLDv
TESTBv
LIS,294v
297 ERROR AAAAAAAAAAAAAAAAAAAAAAAAAAAAAv
298 ENDv
LIS,296v
297v
RUNv
01000v
010v
010v
STOPv
298 ENDv
RUNv
01000v
010v
010v
STUPv
REPLACEv
BYEv
↓
(eor punch)
ANSWERBACv
BOBSIM2v
7744526v
FORv
OLDv
TESTAv
LIS,753v
RUNv
20000v
01000v
.08v
4v
752C AAAAAAAAAAAAAAv
753C BBBBBBBBBBBBBBBv
LIS,752v
REPLACEv
RUNv
20000v
01000v
.08v
4v
BASICv
OLDv
TESTCv
```

Figure 15-6. Stimulator Input File, Example 2

```
LIS,210v
RUNv
400v
REPLACE,TAPE1v
BYEv
↓
(eor punch)
ANSWERBACv
BOBSIM1v
7744526v
BAS,OLD,TESTBv
LNH,294v
297 ERROR AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAv
298 ENDv
297 ERROR BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBv
297 ERROR CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCv
LNH,260v
297v
RUNv
01000v
010v
010v
STOPv
298 ENDv
RUNv
01000v
010v
010v
STOPv
REPLACEv
BYEv
(eor punch)
```

Figure 15-6. Stimulator Input File, Example 2 (continued)

8. Enter TELEX. CR. (If any control point activity is present, enter BLITZ before entering CR TELEX.) TELEX now comes to CTL. PT. 1.

9. See Chapter 4, Section 5, of the Installation Handbook for the description of how to start the Stimulator.

10. To watch TELEX operate, enter TB. CR. This will give you the TELEX display and allow you to monitor the Stimulator activities.

11. Once TELEX has been activated and the Stimulator is running, AUTO can be entered to initiate BATCHIO, and automatic job processing. AUTO must be entered if a BLITZ was entered.

15.2.3 Input Data File

The Input Data file is an item required by the Stimulator. The file must contain all of the information a TELEX user would enter if he were working from a TTY. This includes the user account number, his password, and files and commands he may wish to submit. The file must be in the following format:

1. Data Tape File Identifier (10 Char. Max)
2. Valid User Account Number
3. Valid User Password
4. System (FORTRAN, BASIC, NULL, etc.)
5. File Name (OLD, NEW, LIB)
6. Data (may be a program)
7. Legal TELEX Command (RUN, LIST, SUBMIT, etc.)
8. Data for a Program
9. BYE
10. A telephone Logoff Character (↓) the down-arrow.
11. An EOR

Each and every line of data must be followed by the V symbol. This is the character TELEX recognizes as the TTY carriage return. Each entity of a group of a user's typical data and commands must be followed by an EOR (end of record). This is necessary for the Stimulator to distinguish between each element. The two special punches are:

1. ↓ = 11/8/6 punch - only used for KRONOS V2.0
2. V = 11/0 punch

A further discussion of the Simulator file follows

To create a Stimulator load (test) file, each file must have the following format:

1. 1st card - answerback code (10 characters or less) followed by the down carat (V, 11-0 punches).

2. 2nd card - user account number followed by a down carat.

3. 3rd card - password followed by a down carat.
4. 4th card - system desired, BAS, FOR, EXE; followed by the down carat.
5. 5th card - file type - OLD, NEW, or LIB; followed by a down carat.
6. 6th card - file name followed by a down carat.
7. The body of the program, if a new program. Each card of the program MUST terminate with a down carat.
8. Any command desired, i.e., RUN, SAVE, etc. followed by down carat.

   If data is required for a RUN, the data must follow the RUN command. Each data card must terminate with the down carat.
9. Any other sequences of commands, i.e., LNH, LIST, correction lines or additions to programs, etc., each followed by a down carat.
10. The next to last card must be the BYE command followed by a down carat.
11. The last card must be a down arrow (↓, 11-8-6 punches) for KRONOS V2.0.
12. After the down arrow, put an end-of-record card (7-8-9 punches).

Each program MUST be entered on the load (test) file as an individual record. The Stimulator is capable of pulling off individual records as separate program sequences so that each TTY doesn't have to run the same set of programs. This allows greater flexibility in the stimulated load placed on the system (i.e., it is easier to approach a real world work load). After the last end-of-record card put an end-of-file card (6-7-8-9 punches). All cards must be punched starting in column 1.

15 3 TELEX STIMULATOR SOFTWARE DESCRIPTION

The TELEX STIMULATOR is composed of one CP program, STIMULA, and one PP program 1TS.

When the DSD command STIMULATOR is sensed by DSD, DSD will call 1DS to process the request. 1DS will create an input queue entry for 1TS. 1SJ will initiate 1TS as a subsystem. 1TS will call STIMULA to a CP and prepare to receive session data. STIMULA will read the session file, duplicate and prepare it according to the K display parameters specified by the operator. STIMULA will then build tables which 1TS will use to create stimulation traffic to transmit to 1TD.

Figure 15-7 shows the main flow of STIMULA.

15.4 TRANEX STIMULATOR

The transaction stimulator is composed of two CP programs and one PP program. The user prepares a session file and calls PRESIM via control card to prepossess the session file.

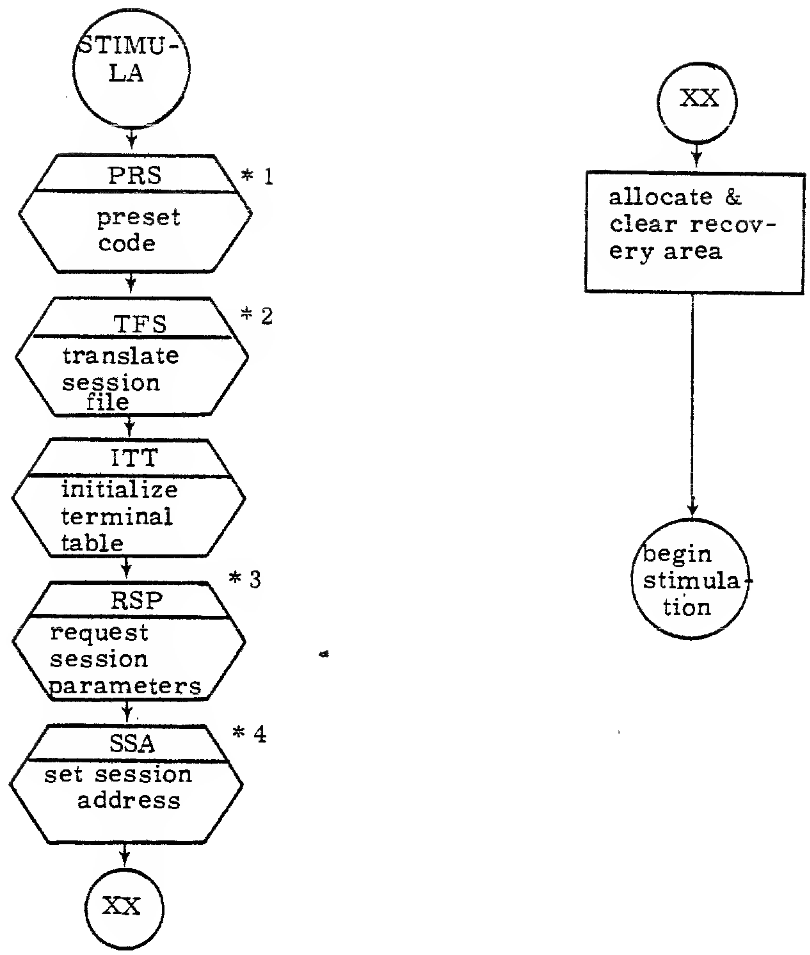


Figure 15-7. STIMULA - Main Program

* 1 Initialize most tables, tell 1TS STIMULA is ready, get session file name, etc.

* 2 Read session file and store in the session buffer.

* 3 Get the parameters (delay time, number of ports, etc.) via the 2nd K display

* 4 Relocate indices for tables.

The user can develop a program which can communicate with TRANSIM and direct the progress of the stimulation. Refer to chapter 9 in the KTS reference manual. The user then calls TRANSIM via control card. TRANSIM will start the PP program CS1; then it will begin processing the restructured session file produced by PRESIM. TRANSIM builds CM tables in its field length which are read by CS1.

CS1 will transmit the sessions to 1TD and will build CM tables for TRANSIM from the responses transmitted back by 1TD.

15.4.1 CS1 Main Loop

When the channel goes active, the function code will be accepted by CS1. If it is not an output, CS1 is out of synch with 1TD. Otherwise, CS1 accepts nl number of words of data, and sets up the pointers in TT, TA, and the CM buffers. Then CS1 processes the output from 1TD and sets up the next input to 1TD.

CS1 waits until the channel goes active again, picks up the function code and verifies that it is an input. If not input, then CS1 is out of synch with 1TD; else transmit the output to 1TD.

The MUX pointer is then set to the next MUX, CS1 pauses for relocation, and then starts the main loop over again.

If CS1 is ever out of synch with 1TD, it will issue the message "OUT OF SYNCH" and terminate the stimulation.

15 4.2 CS1 Internal Descriptions

CS1 - PP portion of the transaction terminal stimulator. CS1 is called by the CP portion of the stimulator and performs all interface functions to 1TD (the normal TELEX terminal driver). CS1 will appear to 1TD as if a real MUX(es) was on the other end of the channel from 1TD. In the case of the non "direct" channel transfer, a special EST entry is required for CS1 to know the channel desired for running the stimulated terminals. The channel number for this EST must be an unused one or the equipments on the channel, 1TD and CS1, will not operate properly. CS1 will get the terminal input from CM buffers handled by the CP program and will put terminal output into CM buffers handled by TRANSIM.

Special EST entry format:

12	12	12	12	12
2000	00 ch	number of lines	2424	60B

where:

ch = channel number to use
2424 = equipment type TT
E = equipment number

For the direct coupled assembly, TELEX does not have to be dropped and CS1 is not as sensitive to what is on the channel. The direct assembly will not function the channel so there can be equipment on the channel while the stimulation is running

logic flow - CS1

1. Initialization
   - Initialize tables.
   - Wait for TELEX and TRANEX to come up

     Note: If channel I/O CS1 waits for TELEX and TRANEX to be dropped

2. Main Loop
   - Check line buffers for available data.
   - Communicate with 1TD.
   - Process output data.

3. End
   - Request TELEX be dropped if channel coupled.
   - Turn off stimulated MUX if channel coupled.
   - Drop control point if an abort caused the end.

15.4.2.1 Line Buffer

CS1 makes use of a series of buffers in the CM field length of the stimulator. One of these buffers will exist for each line on each stimulated MUX. The format of a line buffer is:

1. Word 1

42	6	6	6
terminal name	0	flags	pc

where:

pc - polling code for current terminal

flags -

Bit 11 - 1 when the CPU has put into the buffer an input transaction (Set by CPU).

Bit 10 - 1 when CS1 has passed the message onto TELEX and TELEX has accepted it (set and cleared by CS1). Bit 11 will still be set when this bit is set to 1.

Bit 9 - When no more input for this line (set by CPU) .

Bit 8 - When output available (set by CS1, cleared by CPU).

CPU sets bits only when they are all cleared. The CPU clears them when the transaction is written out.

2. Words 2-15

66
data

where:

data = input transaction to TELEX will be put here. The message will be all display code and will contain a 73B terminator (Set and cleared by the CPU). The message will never be cleared out before another is put in.

3. Word 16

60
time in

where:

time in = time the CPU put the last message into the input buffer (set and cleared by the CPU).

4 Word 17

60
time out

where:

time out = time the first character was placed in the output buffer (set by CS1, cleared by CPU).

5. Words 18-20

60
data

where:

data = output from TELEX is put here in display code (set by CS1, cleared by CPU).

15.4.2.2 TT For CS1

The main control table for CS1 is the TT. There is one TT for each line on each stimulated MUX. Each TT is 5 bytes long The tables are allocated at CS1 preset and the address of the first TT of the group is in the direct cell TT These tables are PP resident.

TT Format consists of the following five bytes.

Byte 0 - used to point into the TA CM assembly/disassembly word.

6	6
char	word

where:

char = next character position to use in the TT. Value is from 0 to 9D, where 0 = left-most character position. This is used to insert and remove characters into/from TT.

word = relative address of current CM word in use in the CM message buffer. This is used when reading a transaction from the CM buffer to give to 1TD, and when transferring output from 1TD to the output buffer. When word = 0, the 1st word of the area is in use.

Byte 1 - Address

12
address

where:

address = address of next routine to use in processing this line.

Byte 2 - Flags and Polling

6	6
Flags	ti

where:

flags = 11, 7, 6

Bit 11 - 1 when the last sequence from 1TD was a select.

Bit 7 - 1 when ETX is received from system and cleared at completion of poll or select (output).

Bit 6 - not used

ti = polling code for this terminal (terminal ID)

Byte 3 and 4 -

3	9
sq	bcc

where:

sq - sequence number for 1TD-CS1 coordination

bcc - block check character (longitudinal parity) All characters sent to 1TD go into the calculation of the bcc except the delay word (see DL parameter in this section and bcc itself).

15.4.2.3 TA Tables

The TA tables are used for assembling/disassembling words from CM. There is one TA table for each line on each stimulated MUX. Each TA is 5 bytes long and is PP resident

The TA tables are set up at CS1 preset and the address of the first TA is contained in the direct cell TA.

When the TA is used for reading the transaction from the CM transaction buffer, the contents of TA will match the contents of the current word of the transaction buffer. When the TA is used for putting the output from a transaction into CM, the TA will be filled with output data one character at a time. When it is full, it will be put into the CM output line buffer. In both cases, the contents of the -word- counter in the TT byte 0 will also point to the current word of the CM buffer.

NIXDORF line protocol:

```
    Sequences received
        poll        - C T B
                      / I C
                      R   C

        select      - C T B                                E
                      / I C - up to 80 data characters -   T
                      R   C                                B

                                  or

                      C T B                                E
                      / I C - up to 80 data characters -   T
                                                           X

    Response sequences:

        to poll
        no data     - C
                      /
                      R

        data        - C T B                 E B
                      / I C - input data -  T C
                                            X C

        to select   - C T B
                      / I C
                      R   C

    Characters:
        C
        /  - command response character
        R
```

```
     Bits 11-8   - for use of MUX
             7   - 0 (indicates not data char.)
           6-5   - sequence number
           4-1   - poll, select, data, no data codes
             0   - parity (lower 8 bits only, odd)
T - terminal identifier (polling code)
I

B
C - block check character (longitudinal parity)
C
Data character
     Bits 11-8   - for use of MUX
             7   - 1 (indicates data character)
           6-1   - character (ASCII)
             0   - parity (on lower 8 bits only)
```

Note

If DEBUG is defined as an assembly constant, a CS1-TRANSIM debug package is assembled.

If DIRECT is defined as an assembly constant, the code to communicate directly with 1TD will be assembled. If not, CS1 talks down the designated channel.

15.5 PRESIM

PRESIM generates input for TRANSIM and in some cases, TELEX and TRANEX. TRANSIM inputs the file containing the reformatted user terminal names and transactions. TELEX and TRANEX input SIMFILE.

Error messages are written on the file OUTPUT.

Control card call:

PRESIM(P1, P2, P3, P4, P5, P6) is defined in Section 9 of the KTS manual.

PRE(N) main processing loop.

Function of routine -

Main program loop processes the session file (I) containing terminal names and transactions.

Format of file (I)

Each logical record contains one or more terminal names each followed by one or more transactions. If two consecutive terminal names are encountered, a diagnostic is written to file (0) for the first terminal name and the second terminal name is assumed valid. Terminal names are verified against the names in the terminal name table (TNT) generated from SIMFILE. Delay and repeat parameters are decoded. The transaction image is not edited. The grouping of terminal names and transactions by logical record generates

an index entry for the corresponding logical record on file (N). Process control is passed to the various subroutines based on the status resulting from the process next card (PNC) routine.

Three conditions are possible -

1. (X1) are not zero, i.e., an EOR/EOF occurred.
2. The (CTE) are not zero, indicating a terminal name card occurred
3. Otherwise, a transaction image occurred.

Entry - (B1) = 1

Exit - None

Routines called - PNC, PIP, WIN, WTR, WIN, READ, RECALL, CLOSE, WRITER, ENDRUN, ERR

Data areas - INC, TTL, TNC, I, PBF, N, O, ATM, TNC

Registers used - A - 1, 2, 4, 6
X - 1, 2, 3, 4, 6, 7
B - None

15 6 TRANSIM

The STIMULATOR provides a means of exercising the system for reliability and Q.A. purposes. System changes can be made and a load created via the stimulator which duplicates live environment usage

TRANSIM cracks the control card (see Section 9 of the KTS manual) and picks the necessary fields from it The following fields are interpreted:

P=lfn	where lfn is the file which contains the user transaction data. The default is NEW.
R=nnn	Run the input nnn times. The default is one
DL=nnn	Delay nnn seconds between inputs (0-600 seconds). The default is 10 seconds.
0=lfn	Output is put on file lfn. If lfn is not designated, the OUTPUT is dropped.
B=lfn	The users program is found on lfn in absolute program format The default for lfn is LGO.
D -	The D option allows the application programmer to debug his program without going to the TELEX system with the input. The program will not call CS1. It is up to the user to set the write bit if he wants to write the transaction out after processing it. If the input ready flag is not cleared, control will not return to the user on the line. If no load file (B) is designated, the transactions are copied to the output area and set ready to be written to the OUTPUT file. If there is no OUTPUT file, the transaction is not copied.

NR -	Don't rewind the output file before writing on it. The scratch file is rewound unconditionally.
LN-nnn	Number of lines is equal to nnn.

The input file for the STIMULATOR is card image format. The transaction should appear exactly as an operator would type at the terminal. Terminal function keys (unlock keyboard, send, etc.) are not represented in the format.

User Interface - Common area USR for communication between user program and TRANSIM.

Word	Meaning
1	=1 - preview, =0 - review
2	time delay in seconds
3	repeat count
4	current buffer number
5-n	random address for each line (total of n lines)
n-m	input and output buffers

Output Format

42	12	6
terminal name	flags	polling code

where:

Flags 11, 10, 9, 8

Bit	Meaning
11	input for CS1
10	input has been sent
9	EOR - no more input available for this line
8	output available

input data area (13 words)
time of input
time of output was received
output data (62 words)

Time of Input and Output Format

24	18	18
hour	minute	second

Random Index/TRANSIM Next-Read Table

Current terminal name

14	21	7	7	30
delay time	status	out address	in address	random address

The table is two words long for each line.

Data Format

	12	6
terminal name	0	pc

where:
pc = polling code

followed by data terminated by a zero byte

TRANSIM-CS1 central memory interface

Word	Byte	Meaning
10-(SCMP)	0	length of a line buffer and output on or off bit (59)
	3-4	address of 1st line buffer
14-(ESTI)	4	0 unless run to be aborted
15-(CMMN)	0	20MM MM=starting MUX number
	4	1 for recall
16-(CMWW)	4	0 until POLTBL construction
11-13		CS1 debugging aids
17-21		CS1 debugging aids

15 6.1 Logic Flow For TRANSIM

1. Initialization
   - Crack control card parameters
   - Setup CS1 communication area
   - Call CS1
   - Preset line buffers
   - Read random file index to USR (for user program) common area
   - Setup USR (for user program) common area
   - Transfer poll table to CS1

2. Main Loop
   - Process each line to put data into its buffer
   - Process USR own-code entrance - preview
   - Process output data
   - Process USR own-code entrance - review

3. End
   - Complete all active files
   - Put out correct dayfile messages

15.6.2 TRANSIM Messages

1. COPYING DATA TO THE OUTPUT FILE

   The output data is written to a scratch file as one large logical record. This scratch file is reblocked to the user designated OUTPUT file.

2. STIMULATION ENDED

   Stimulation input has been exhausted; all output has been received from 1TD

3. SUB-CONTROL POINT ERROR

   The user has committed a fatal sub-control point error TRANSIM issues the message and aborts.

4. INPUT IS MISSING RANDOM INDEX

   The input file does not have a legal random index as its last record

   TRANSIM load error-premature end of load file. An end of record was encountered before the identification tables had ended. Also, an unrecognized table may have been encountered.

5. TRANSIM LOAD ERROR-COMMON AREA TOO SHORT

   The USR defined area is not large enough for this run.

6. TOO MANY TERMINALS

   The number of terminals designated to be stimulated is greater than the number that TRANSIM is assembled for.

7. ERROR IN TRANSIM ARGUMENTS

   The control card parameter is bad. TRANSIM issues the message and aborts

8. MAXIMUM NUMBER OF LINES ARE XXX

   There are XXX lines of data available

9. REPEATING INPUT

   Each time the input is repeated, this message is put out

EXPORT/IMPORT 16

16.0 INTRODUCTION

The EXPORT/IMPORT subsystem coordinates communication between Control Data 6000 Series or CYBER 70 computer systems and remotely located 200 User Terminals.

16.1 EXPORT/IMPORT PROGRAMS

The EXPORT/IMPORT subsystem consists of the following programs:

- E200 CP – CPU Program
- 1LS – PP Program – transient
- 1ED – PP Program – dedicated
- XSP – PP Program – transient

16.1.1 E200 CP is a CPU program which reformats input/output data of the 200 User Terminals. Its Field Length (FL) is also used for all the communication tables and FETs for the subsystem. The common deck, COMSEXP, is used to establish the constants, pointers, and communication table areas. Figure 16-1 shows the layout of these areas.

A local RPL for the 1LS overlays is contained in the E200CP area. The FETs and buffers are kept in the upper portion of the FL, so that they may expand and contract as the need arises. Figure 16-2 shows the E200CP core layout and its interaction with the three other routines of this subsystem. It also illustrates the subsystem interaction.

16.1.2 1LS is a transient PP program which processes and assigns files, performs functions for 1ED, and is the EXECUTIVE routine for the subsystem.

16.1.3 1ED is a dedicated PP program which controls communication between the CDC 6000 Series or CYBER 70 computer systems and the 200 User Terminals. 1ED must get E200CP out of auto-recall when there is input to process. 1ED must convert the output to 200 User Terminal format and output to that terminal. 1ED is the only program constantly running in the subsystem and polling the 6671 multiplexer.

16.1.4 XSP is a transient PP program which is called by 1LS to perform time-consuming tasks.

Address Relative to RA	Octal 0000	System Communication Area	Number of Entries	Number of Words per Entry
DRCL	0100	E200CP auto recall word set by 1ED when some activity is needed		
	0103			
FWTL	0104	TFS – Function/Status Table	NT·TFS=2	NT·TFS*N·PORTS*
	0144	MSGB – Message Buffer	NT·MSGB=4	NT·MSGB*N·PORTS
	0244	LINF – Login Information Table	NT·LINF=2	NT·LINF*N·PORTS
	0304	CPIK – CPU Interloc Table	NT·CPIK=1	NT·CPIK*N·PORTS
	0324	DPJT – Drop Job Table PWLT – Password Table for VUN	NT·DPJT=1 NT·PWLT=1	NT·DPJT*N·PORTS NT·PWLT*N·PORTS
	0344	JST – Job Statists Table	NT·JST=1	NT·JST*N·PORTS
	0364	FAMT – Family Name Table	NT·FAMT=1	NT·FAMT*N·PORTS
W·INT=	0404	1LS recall word auto recall until 1LS is ready from call via TLX when DRCL word went non-zero by 1ED		
	0405	Initialization Flag		
	0406	Scanner Position		
		The 8 common routines defined in Paragraph 16.4		
	0720	E200CP Code		

*N·PORTS=16D maximum number of ports in a 6671.

Figure 16-1. EXPORT Communication Areas

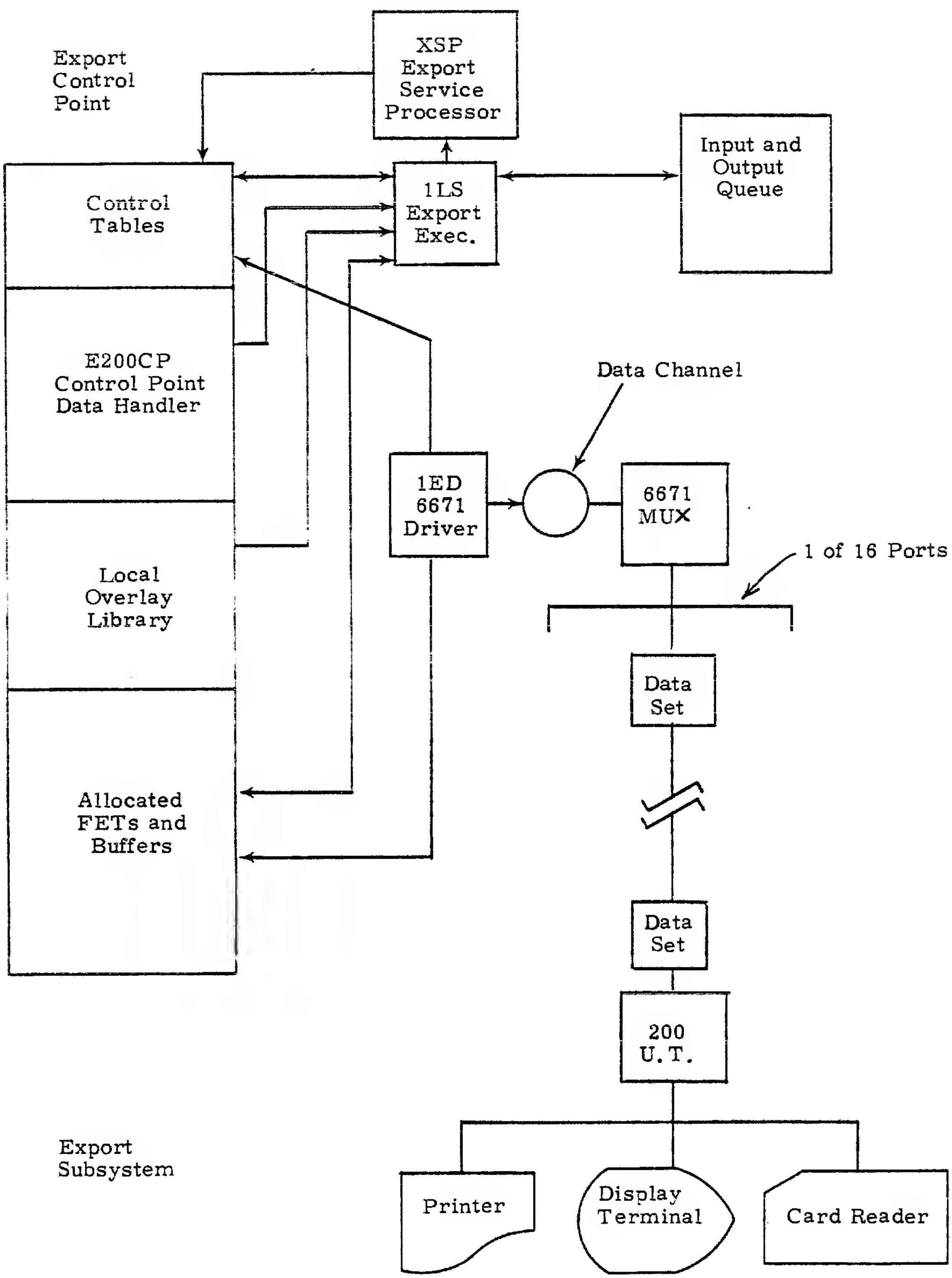


Figure 16-2. E200CP Layout and Interaction

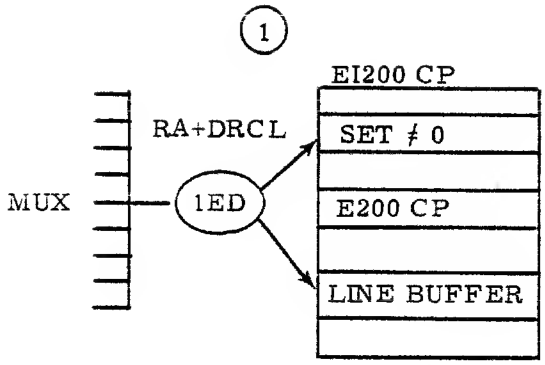
16.2 EI200 OVERVIEW

Logically, but not physically, Table 16-1 is maintained for each port (16 per mux). Each area of the table is detailed in paragraph 16.3. Figure 16-3 illustrates the sequence of operations and data flow of EI200.

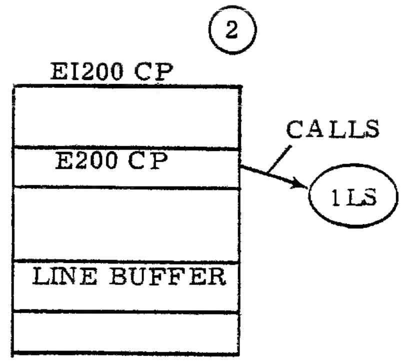
TABLE 16-1. PORT-MULTIPLEXERS MEMORY

	12	12	12	24
FUNCTION WORD	FUNCTION 1ED TO 1LS	TERMINAL ID	MUX EQ NO.	NOT USED
STATUS WORD	CP I/O STATUS	I/O DRIVE	INPUT FET ADDR	OUTPUT FET ADDR
MESSAGE BUFFER	Messages To/From Remote Display Screen			
LOGIN INFO TABLE LINF	Display Code User Number			1
	Jobname		UI	STATUS
CPU INTERLOCK TABLE CPIK			Input Active	Output Active
DPJT And PWLT	Internal System Jobname			Response *1
FAMT	Family Name			
OUTPUT FET 8 WORDS	Internal System Name			STATUS FIRST IN OUT LIMIT
BUFFER				
INPUT FET 8 WORDS	Internal System Name			STATUS FIRST IN OUT LIMIT
BUFFER				
LINE BUFFER 8 WORDS (1 Card Image)	Input from Card Reader comes here via 1ED (1 Card Image at a time)			

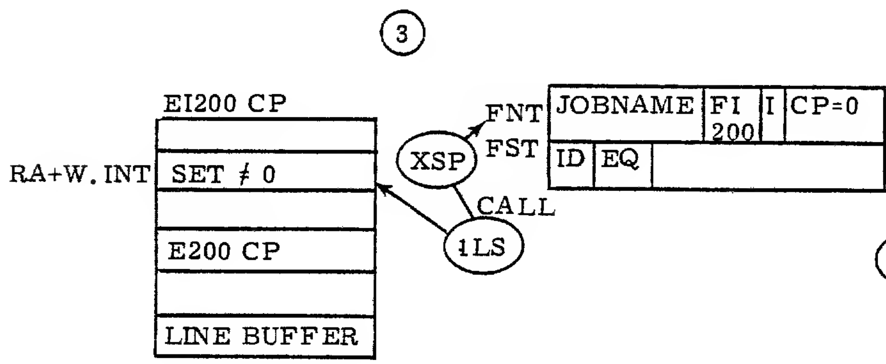
* Contains PASSWORD of log in user at log in time until user verified. Used for dropping jobs as long as user is connected.



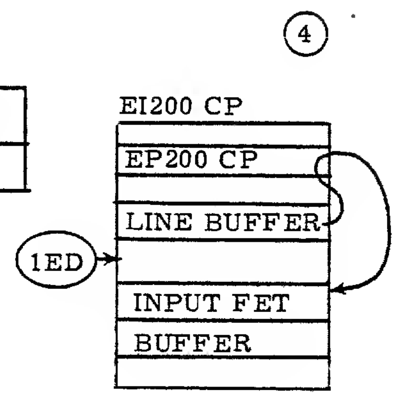
1ED reads from MUX to line buffer and sets RA+DRCL words non-zero. This will take E200 CP out of Auto-Recall.



E200 CP calls 1LS and goes into Auto-Recall.



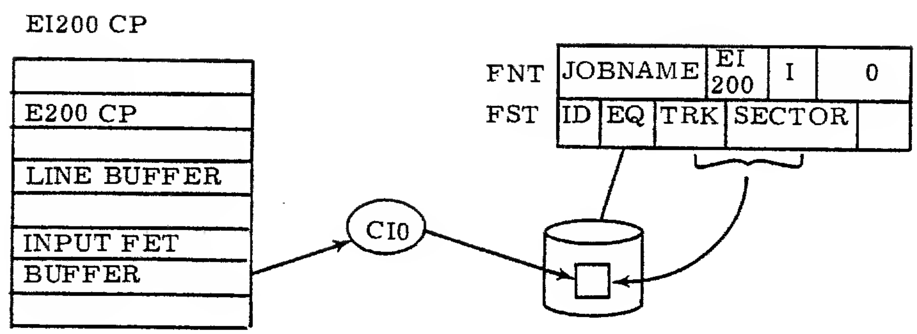
1LS sets RA+W·INT non-zero, which will take E200 CP out of Auto-Recall. 1LS calls XSP to create an FNT/FST input queue entry for the job in the Line Buffer. Using 0BF and 2TJ (to crack the job card).



E200 CP reformats the Line Buffer data as 1ED passes it and moves the data to the Input FET Buffer.

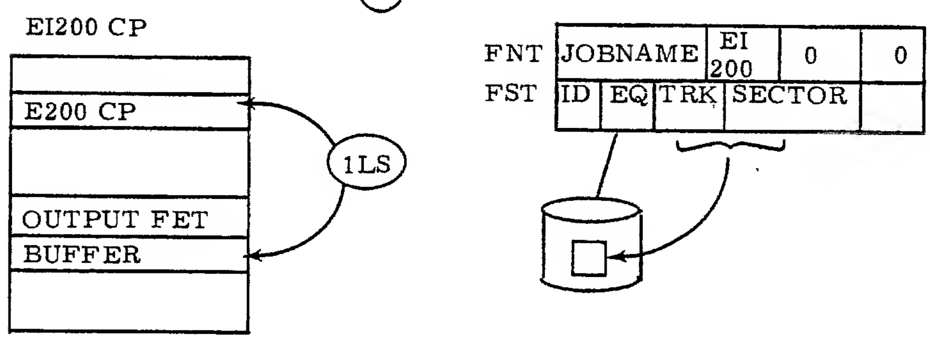
Figure 16-3. EI200 Overview

5



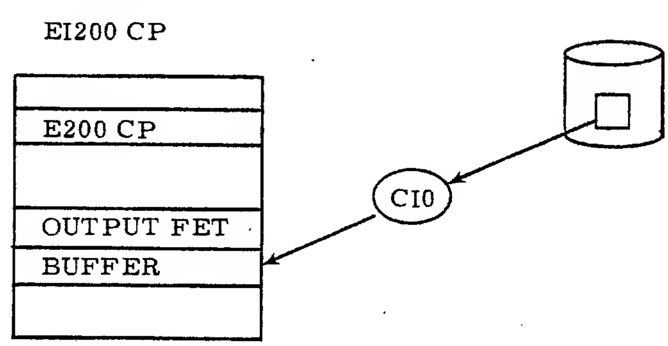
E200 CP calls CIO to write the data from the Input FET Buffer to the disk.

6



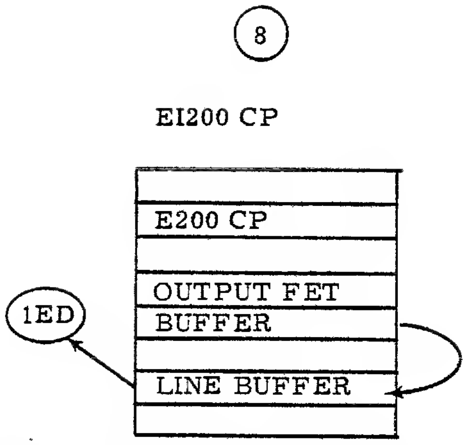
1LS finds an output queue entry and creates a banner page in the Output FET Buffer and informs E200 CP.

7

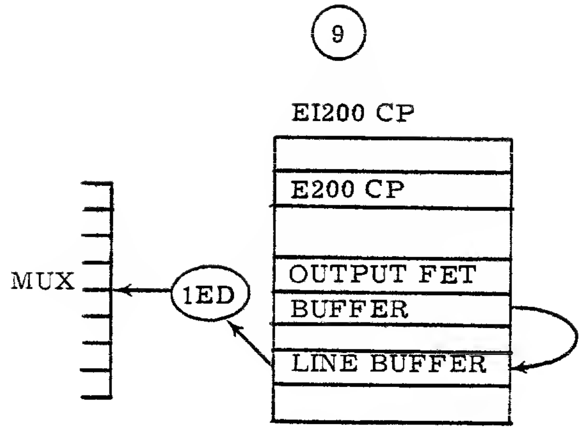


E200 CP reads the Output File via CIO into the Output FET Buffer.

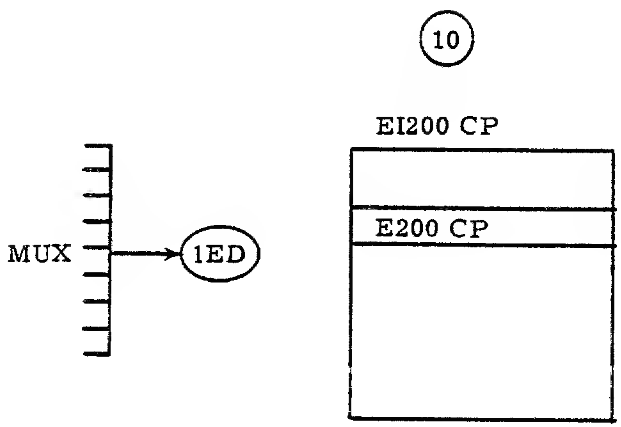
Figure 16-3. EI200 Overview (Continued)



E200 CP formats the Output FET Buffer for the remote printer and informs 1ED.



1ED moves the Line Buffer data, one line at a time, to the remote printer.



E200 CP goes into Auto-Recall and 1ED continues to poll the MUX.

Figure 16-3. EI200 Overview (Continued)

16.3 EXPORT COMMUNICATIONS AREAS

The following functions are processed by 1LS for 1ED. These tables are in the E200CP FL and are used for communication areas for all parts of the subsystem.

- TFS – Function Table
- TFS – Status Table
- MSGB – Message Buffer
- LINF – Log-in Information Table
- CPIK – CPU Interlock Table
- DPJT – Drop Job Table
- PWLT – Password Table
- FAMT – Family Name Table

16.3.1 Function Table TFS

12	12	12	12	12
FUNCTION TO EXEC	TERMINAL IDENT	MUX EQUIP-MENT NUMBER	NOT	USED

Equipment number of MUX assigned
This field used in ENTRY 0 only.

Site address of logged in terminal.

Set by I/O driver (1ED) to communicate with EXEC (1LS).

These are defined in 1LS's field length and also in COMSEXP.

- 00 - Null function
- 02 - Message from terminal
- 04 - Print block complete
- 06 - Special end read
- 10 - Write message complete
- 12 - MUX not available
- 14 - MUX not operational
- 16 - Initialization complete
- 20 - Terminal connected
- 22 - Printer not ready
- 24 - Message read error
- 26 - Terminal disconnected
- 30 - Operator interrupt
- 32 - Read E3, no EOF
- 34 - Read E3, with EOF
- 36 - Read E2, no EOF
- 40 - Read E2, with EOF

16.3.2 Status Table TFS

12	12	18	18
CP I/O STATUS	I/O DRIVE	INPUT FET ADDRESS	OUTPUT FET ADDRESS

1LS responds to 1ED.

1LS responds to E200CP.

Note

E1, E2, E3 are hardware functions set by both the remote card reader and printer. They are specified in the appropriate EI200 hardware manual.

CP I/O STATUS –
- 0 - Run CP. (output, coded mode)
- 1 - Run CP. (input, coded mode)
- 2, 3, 4 - not assigned
- 5 - Return sequence number
- 6 - Output file active
- 7 - Input file active
- 8 - not.assigned
- 9 - Output file suspended
- 10 - Read, wait for operator "GO"
- 11 - 0 = Read E3, 1 = Read E2 on previous read

I/O DRIVE –
- 0 - Terminal on line
- 1 - Terminal logged in
- 2 - Interrupt during print transmission
- 3 - Interrupt during read transmission
- 4 - Wait for storage
- 5 - Not assigned
- 6 - Execute print control program
- 7 - Execute read control program
- 8 - Execute write message to terminal screen
- 9, 10, 11 - Not assigned

INPUT FET ADDRESS – Relative CM address of input file FET

OUTPUT FET ADDRESS – Relative CM address of output file FET

16.3.3 MSGB – Message Buffer

Each message buffer is 4 CM words long. The messages to/from the remote terminals are placed in the appropriate message buffer with a 0000 termination byte.

16.3.4 LINF – Log-in Information Table (2 words/terminal)

This table is used by XSP to respond to 1LS.

42		18
DISPLAY CODED USER NUMBER		ALWAYS = 1
JOB NAME	USER INDEX	STATUS
24	24	12

STATUS — 0 - Log-in active
1 - Log-in complete
2 - Request PP again (system busy)
3 - Duplicate User Number

USER INDEX – 0 ⟶ Illegal user number

16.3.5 CPIK – CPU Interlock Table

	12	12
	INPUT ACTIVE	OUTPUT ACTIVE

E200CP sets the proper byte ≠ 0 when active on a file and zeros the proper byte when it detects the CPU drive bit off for the appropriate channel (INPUT or OUTPUT).

16.3.6 DPJT – Drop Job Table

42	18
INTERNAL SYSTEM JOB NAME	RESPONSE STATUS

RESPONSE STATUS – 0 - Drop active
1 - Drop completed successfully
2 - PP not available (system busy)
3 - Job name not found
5 - Job found but not dropped

16.3.7 PWLT – Password Table (same location as DPJT)

At log-in time, this table is used for the user password instead of JOB DROP.

16.3.8 FAMT – Family Name Table

At log-in time, this word is used for the user's family name.

16.4 EXPORT/IMPORT FET

The EXPORT/IMPORT FET is created for each terminal logged in. The formats are shown in Figure 6-4.

16.5 EXPORT SYSTEM CENTRAL PROCESSOR PROGRAM (E200CP)

The central processor portion of the EXPORT system is used for the processing of data to and from the remote site.

Data being received from the remote site card reader is placed in the line buffer allocated to the active terminal by the I/O driver program. Very little processing of the received data is performed by the I/O driver itself 1ED. The data is converted to display code and written, one card image at a time, into the line buffer. When the I/O driver senses an end of message code, the CM line buffer is marked full so that E200CP will process that data at the next opportunity. Trailing blank suppression and detection of end-of-record and end-of-file is accomplished by E200CP. Transmission of data to the system allocatable mass storage device is also requested by the E200CP.

The buffer space for an output file is allocated by the executive program 1LS. The banner page data is placed in the circular buffer by the executive program 1LS. All subsequent I/O requests are issued by the E200CP program. Data from the circular buffer is compressed according to the 200 User Terminal specification and placed into the line buffer for transmission to the terminal. As much data as possible is placed in the line buffer on each cycle. A full line buffer is not always possible to generate because the print line cannot be split between messages.

The control for the CP program is a switched circular scan of the terminal control table. Switching is performed by the executive via the status word in the function/status table. Control interlock is through the CPIK table within CM. Every complete scan will attempt to complete an entire operation on every active terminal. When an entire scan is completed, the CP is place stopped – to be restarted by the executive (i.e., E200CP goes into auto-recall).

INPUT FET

<table>
<tr><td>FET +0</td><td colspan="4">Internal System Name</td><td>Code/Status</td></tr>
<tr><td>1</td><td colspan="4"></td><td>First</td></tr>
<tr><td>2</td><td colspan="4"></td><td>In</td></tr>
<tr><td>3</td><td colspan="4"></td><td>Out</td></tr>
<tr><td>4</td><td>FNT Address</td><td colspan="2">0</td><td>0</td><td>Limit</td></tr>
<tr><td>5</td><td>Full/Empty Driver Flag</td><td>Job Card Processing in Progress</td><td colspan="2">Address of Line Following EOR</td><td>Address of Line Following EOF</td></tr>
<tr><td>6</td><td colspan="2">Job Sequence Number</td><td colspan="2">0</td><td>Pointer to Next Allocated FET</td></tr>
<tr><td>7</td><td>Job Priority</td><td>Job Time Limit</td><td>Job FL</td><td>0</td><td>Card Count</td></tr>
</table>

OUTPUT FET

<table>
<tr><td>FET +0</td><td colspan="5">Internal System Name</td><td>Code/Status</td></tr>
<tr><td>1</td><td colspan="4"></td><td>01</td><td>First</td></tr>
<tr><td>2</td><td colspan="5">0</td><td>In</td></tr>
<tr><td>3</td><td colspan="5">0</td><td>Out</td></tr>
<tr><td>4</td><td>FNT Address</td><td>Day File First Tract</td><td colspan="2">Day File First Sector</td><td>0</td><td>Limit</td></tr>
<tr><td>5</td><td>Full/Empty Driver Flag</td><td colspan="5">0</td></tr>
<tr><td>6</td><td colspan="3">Job Sequence Number</td><td colspan="2">0</td><td>Pointer to Next Allocated FET</td></tr>
<tr><td>7</td><td colspan="6">Print Line Count</td></tr>
</table>

Figure 6-4. EXPORT INPUT/OUTPUT FETS

The eight common decks used by E200CP are loaded in the control table section. They are listed in Table 16-2.

TABLE 16-2. E200CP COMMON DECKS

Title	Description
COMCCIO	I/O function processor
COMCRDC	Read coded line, "C" format
COMCWTH	Write coded line, "H" format
COMCSYS	Process system request
COMCRDW	Read words to working buffer
COMCWTW	Write words from working buffer
COMCMAC	CPU system macros
COMCCPM	Control point manager processor

Figure 16-5 is a flowchart of the main scanner control.

16.5.1 INP – Input Data Processor

The following functions are performed by this program:

1. Move data from the line buffer into the file circular buffer, removing trailing blanks in the process.

2. Write data to the system mass storage device using CIO and standard I/O techniques.

3. Sense and process end-of-records. An EOR is indicated by a block of eight words in the line buffer containing the character K·EOR (=30B) (defined in COMSEXP) (12-bit field) in byte zero of block word zero. Issue a CIO request to write EOR from the buffer. If the first word of the next block does not contain EOM (=00B, end of message) (see step 5), set the beginning address of the next block in FET+5, bits 18-35, and continue processing when the FET becomes free.

4. Sense and process end-of-file. An EOF is indicated by a block of eight words in the line buffer containing the character K·EOF (=27B) in byte zero of block word zero. If the word following this eight-word block does not contain an EOM code (=0 end of message) (see step 5), record the beginning address of the next unprocessed data

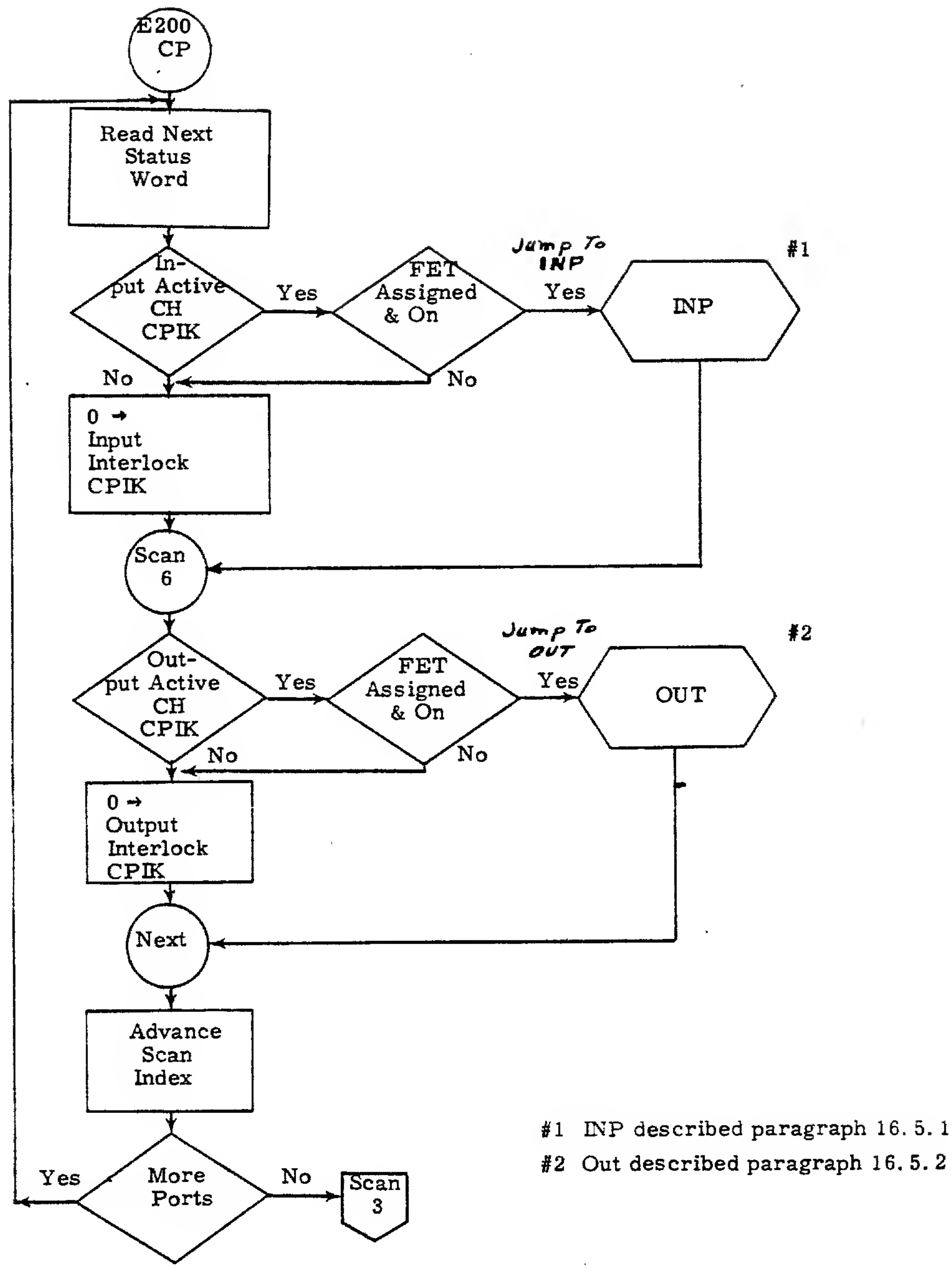
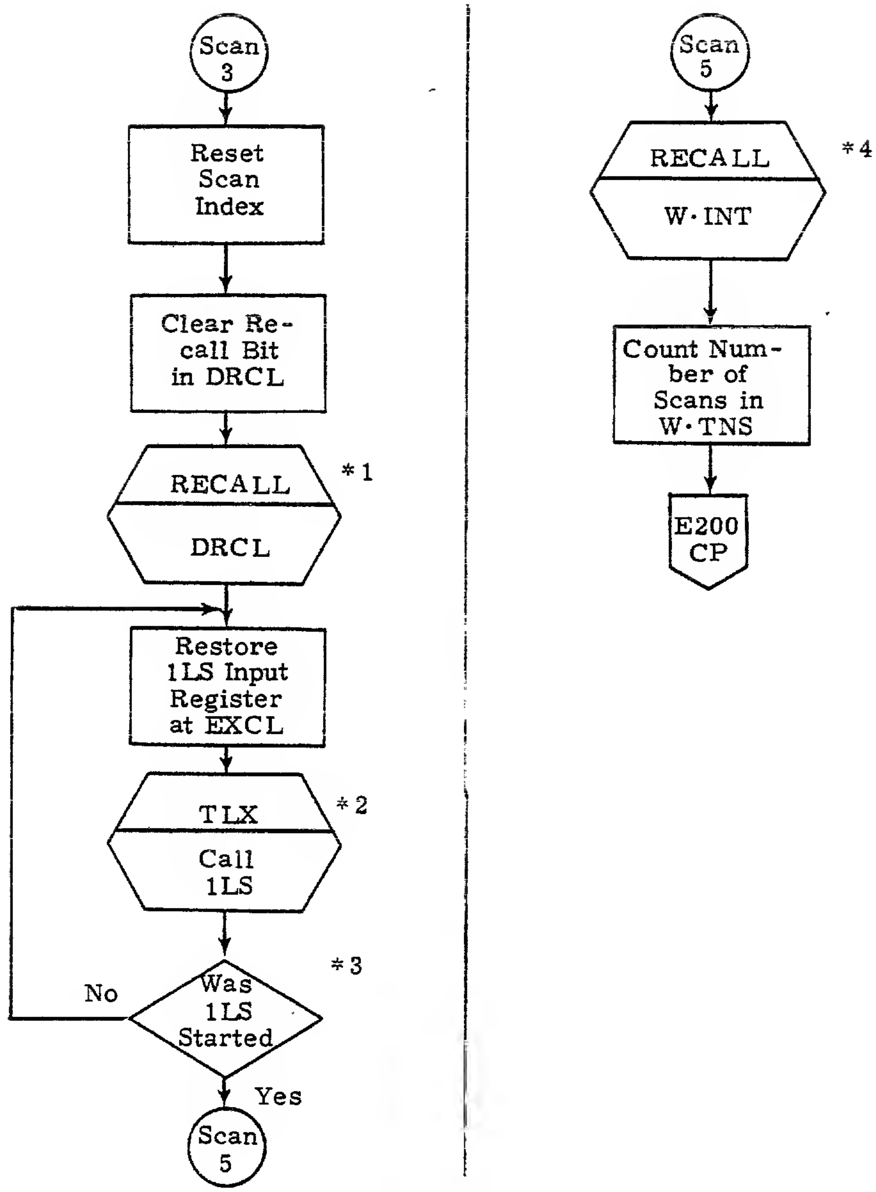


Figure 16-5. E200CP – Control Scanner



*1Place E200CP in auto recall unit DRCL bit 0 is set non-zero by 1ED.
*2Use system RA+1 request TLX to call 1LS.
*3See description of TLX request Section 6 (SEP).
*4Auto recall unit 1LS set W·INT non-zero.

Figure 16-5. E200CP – Control Scanner (Continued)

block in FET+5, bits 0-17, set byte one non-zero, and do not alter byte zero (full/empty control). The CP program will wait for FET+5, byte one to be set back to zero by 1LS when it has processed the input file. Processing of data will then continue at the block address stored in FET+5, bits 0-17.

5. Sense and process end-of-message. An EOM is indicated by byte zero of a block (or special last word) containing the character K·EOM (=0) in byte zero. The full/empty status (byte zero) of FET+5 is set empty and normal data processing continues.

   These special values are:

   K·EOR = 0030B
   K·EOF = 0027B
   K·EOM = 0000B
   K·EOI = 0055B

   They are specified in the COMSEXP common deck.

16.5.2 OUT – Output File Processor

Data from the circular buffer is placed into the line buffer by this phase of the E200CP program.

Strings of blanks greater than two characters in length and up to MAXB characters are replaced by a two-character compression set. Strings greater than the maximum length are processed as one or more strings of maximum length and a remaining short string if necessary. End-of-line codes are placed on every line sent to the remote printer. Only complete lines will be placed in the line buffer and lines of more than 136 characters will be treated as more than one line, but some characters may be lost from lines of excessive length.

An attempt is always made to fill the line buffer with the maximum number of characters allowed. A restriction of the terminal hardware forces a full line to be transmitted before an end-of-message. This means that not all transmissions will be maximum length.

The 200 user terminal has three buffers, the screen, card reader and printer. The screen buffer is used for transmission to the MUX consequently card images are transferred from the card reader buffer to the screen buffer for transmission to the MUX. Similarly, output is transmitted by the MUX to the screen buffer, and is then transferred to the printer buffer for printing.

16.6 1LS – EXPORT EXECUTIVE ROUTINE

E200CP will wait in auto-recall until 1ED sets RA+DRCL to 1, indicating some input was received from some remote terminal. E200CP will call 1LS to a PP and go into auto-recall until 1LS is ready for E200CP to begin processing the input or output.

1LS may load any of the following overlays at anytime, depending on the action required.

1. Initial load – 1LS (loaded by system) loads executive subroutines at 7000B. These two segments are expected to be resident at all times (1LS and 0VRS in core).

2. Function processing – The function processor segment is loaded if any outstanding functions from the driver are found (1LS, 0VRS, and 0VFP in core).

3. Input file processing – The enter queue segment is loaded if function processing found any outstanding input activity (1LS, 0VRS, 0VPJ, and possibly 2TJ in core).

4. Search for and initiate output – The FNT search segment is loaded if the time interval for FNT search is satisfied (1LS, 0VRS, 0VFA and possibly 2LD in core).

5. Storage management – The storage manager is loaded if the time interval for buffer check is satisfied (1LS, 0VRS and 0VCS in core).

Any number of the preceding actions could occur during an executive pass.

The EST entry is expected to be type 3000 equipment type ST. Change MUXDT EQU in the I/O driver if a different equipment type code is desired.

The EST entry is located by the MUX I/O driver program. The search will find the first entry of the proper type which is not set off or assigned to another control point. The EST format is as follows:

7	5	6	6	6	6	12	3	3	6
0	CP number	0	channel number	0	0	Device type	0	EQ num	0

All of the normally used EXECUTIVE overlays are stored in central memory within the field length of the EXPORT program during initialization. This technique was selected to increase the load speed of the PP EXECUTIVE without using large amounts of CMR space if EXPORT is not loaded. For this reason, the programs and overlays associated with EXPORT should be disk resident. The only part of EXPORT that must be CM-resident is the short executive main program, 1LS.

The local RPL map is identical in format with the SYSTEM RPL. Starting at the address in pointer word P·RPG, a zero word ends the library.

The routines in the library are:

1. 9IA overlay OVFP – function processor, when 1ED talks to 1LS via the TFS table.

2. 9IB overlay OVFA – file name table search for print files. It searches the FNT for files to be printed at the remote sites. If any such files are found, a buffer is allocated and the header information is placed in the buffer for the initial print operations. Subsequent data handling is performed by the central processor program associated with this system. It calls overlay 2LD to generate the banner page.

3. 9IC overlay OVPJ – job card processor. It is called by executive main control when needed to process job cards read from the remote terminals and enter complete job files into the input queue. It calls system program, 2TJ, to process the job card.

4. 9ID overlay OVCS – central memory manager. The storage manager executes every few seconds in an attempt to reduce the amount of storage used by EXPORT central memory.

5. 9IE overlay OVIN – initialize EXPORT. The first time 1LS is called by E200CP, this overlay will initialize all of EXPORT.

6. 9IF overlay OVAB – abort EXPORT. All error modes, operator STOP, and error messages are processed by this overlay.

7. 9IG – overlay OVRO – initialize local RPL. Initialize resident library programs in control point FL area. Programs are stored in the same format as RPL system programs. Pointer P·RPG holds the address where this library begins.

8. 9IH – overlay OVRS – resident subroutines. The subroutines are used by the main segment and are loaded into the upper portion of PP memory to allow for expansion of the main segment or any other overlay.

In addition, the two system overlays, 2TJ and 3BB (from BATCHIO), are used. Also, 1LS will call the following system programs:

1. 0DF – Drop files
2. 1AJ – Job advancer
3. 1DL – Display overlay loader
4. CIO – Combined I/O
5. XSP – EXPORT service processor

Figure 16-6 shows the 1LS core layout.. Figures 16-7 and 16-8 are flowcharts of the 1LS main flow.

16.7 EXPORT SERVICE PROCESSOR (XSP)

This program is called by the EXPORT executive (1LS) to assist in certain functions that require more time or space than are available for individual processing tasks within the executive.

The following functions are available:

DJP
VUN
MJE

16.7.1 Process Job Drop Requests (DJP).

	18	6	12	12	12
IR=	XSP	CP#	0	Index into DPJT Table	Function Code for V·DPJT=1

The job name table within the EXPORT CM table area is used for job dropping. At the completion or abortive attempt to drop a job (a user job at a remote terminal), the return status (DPJT + index) is set as follows:

1 – if job dropped
3 – if job not located
5 – if job is located but not dropped

DJP will get the job name from the DPJT + index, will attempt to locate the job in either the system input/output queues or at a control point, and attempt to drop it. If the job type is not EIOT, no action is taken.

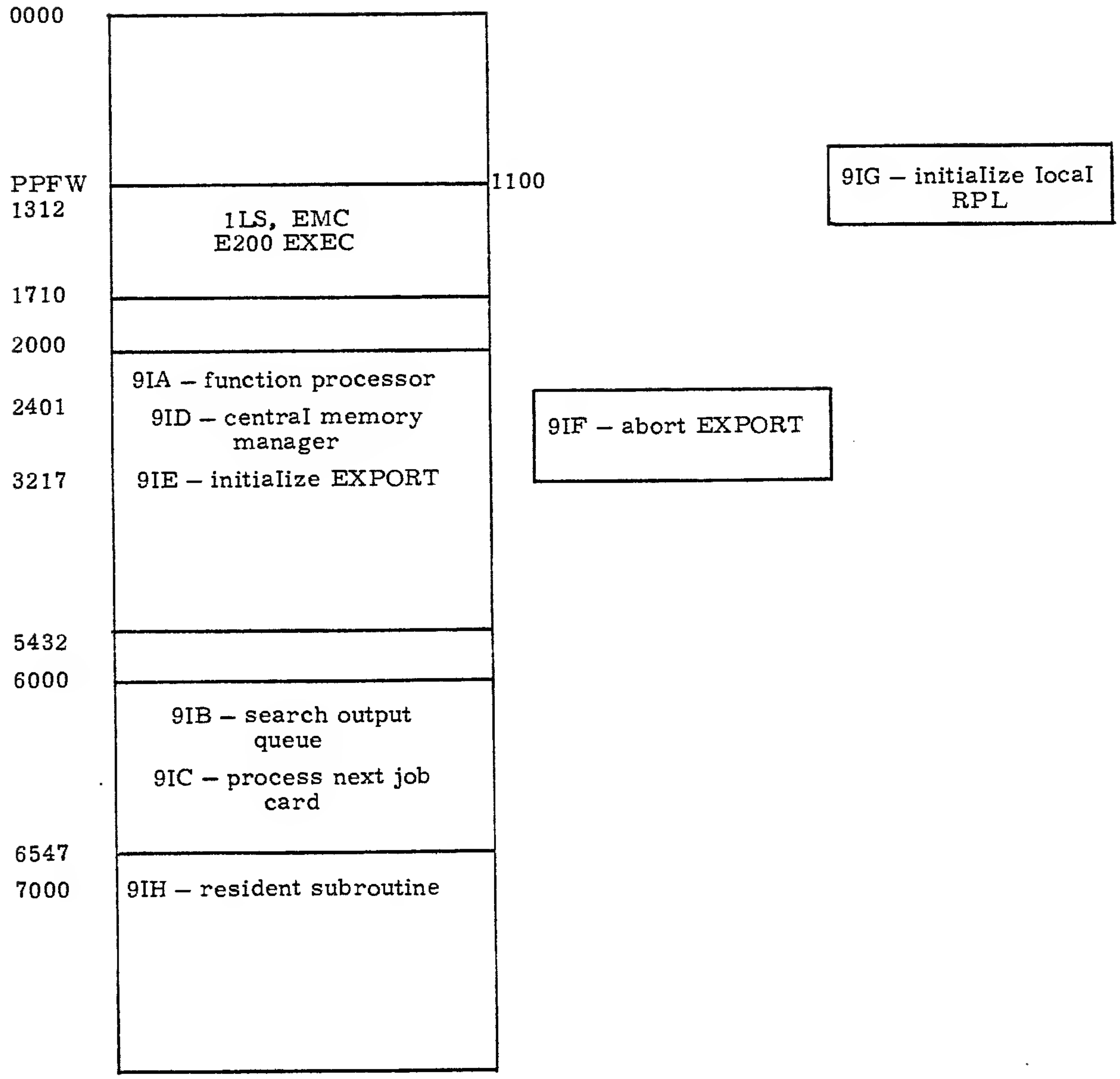


Figure 16-6. 1LS Core Layout

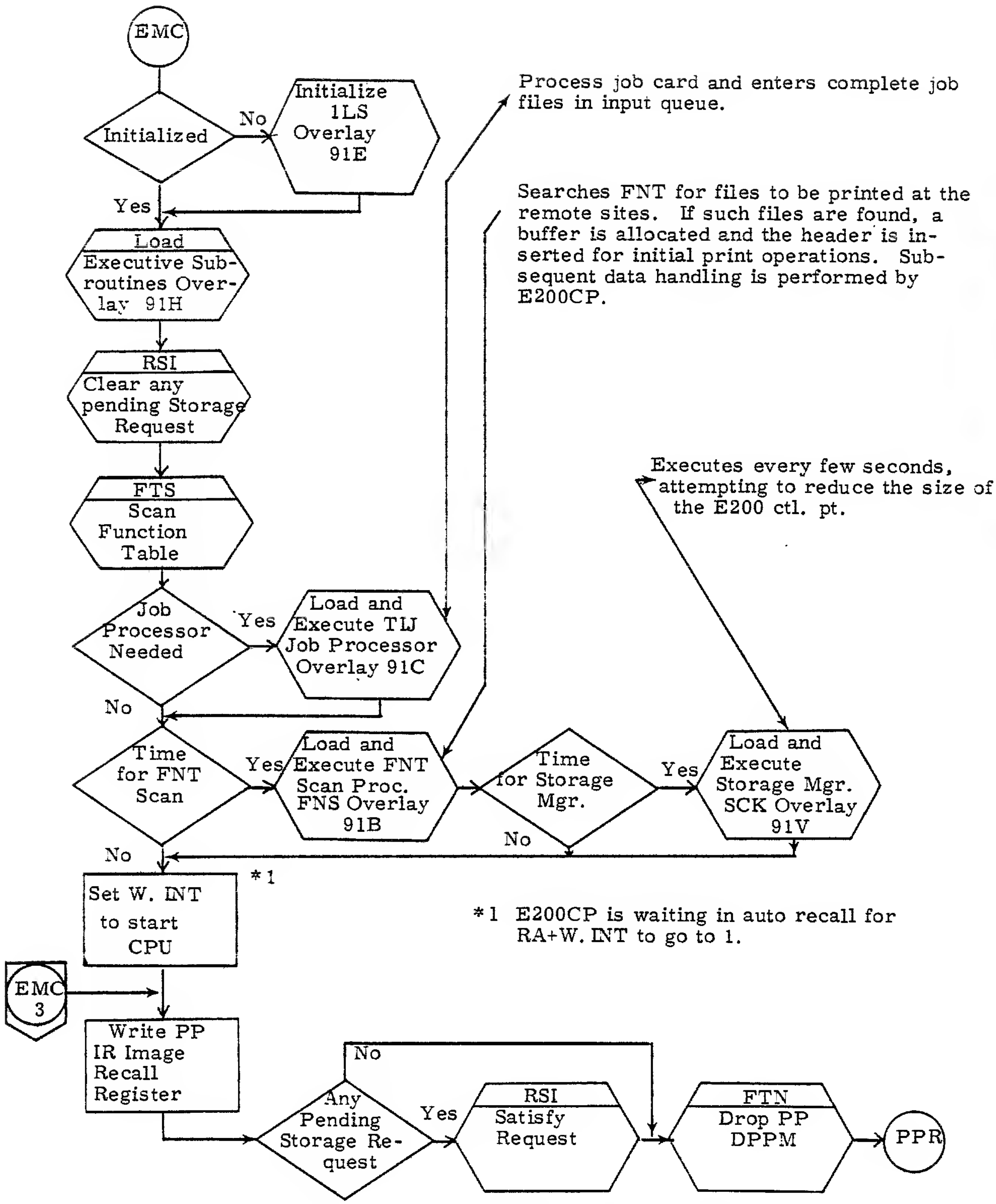


Figure 16-7. 1LS – EXPORT Executive Main Control

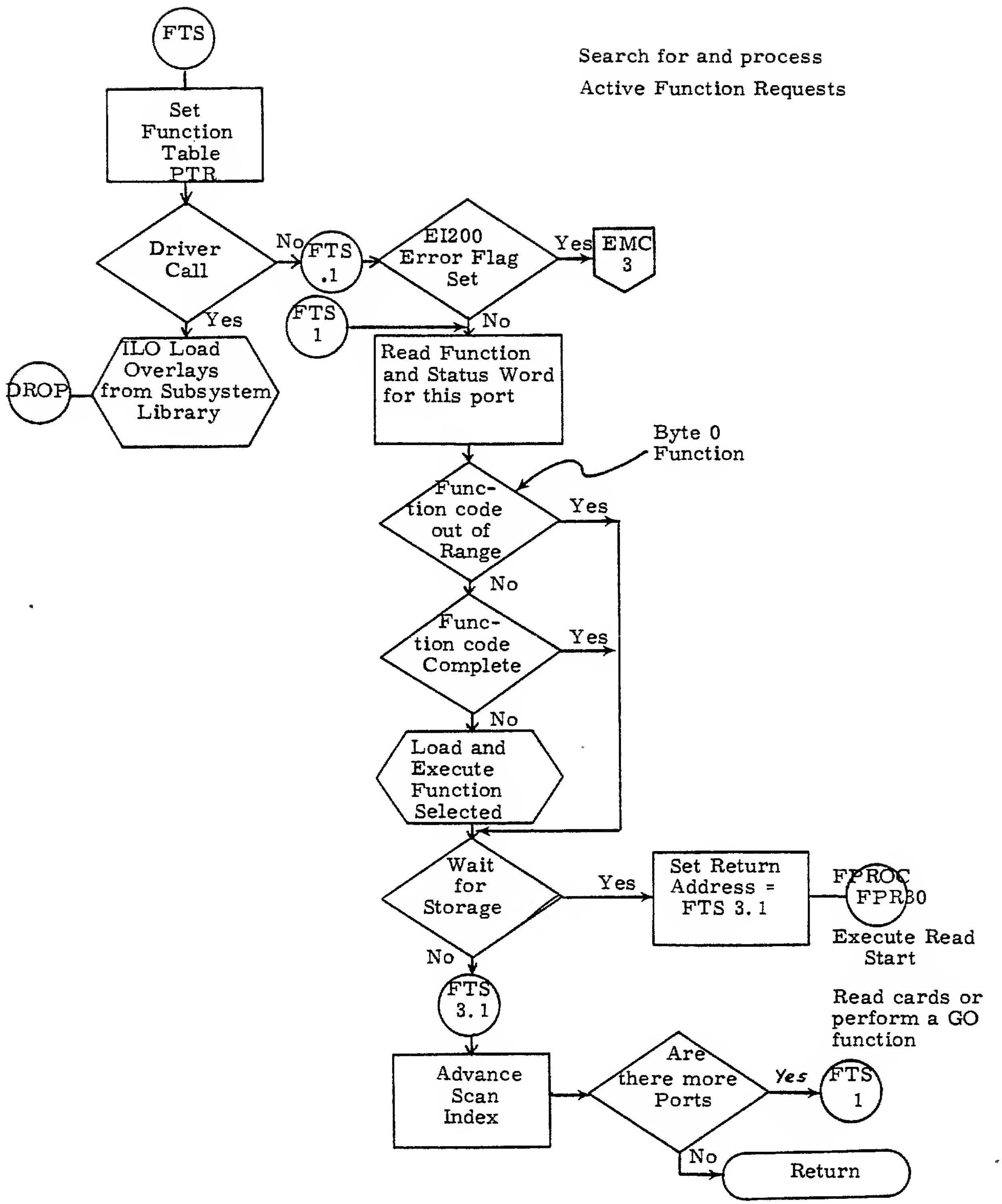


Figure 16-8. Function Table Processor

16.7.2 Log-In Terminal (VUN)

	18	6	12	12	12
IR =	XSP	CP#	0	Index into LINF Table	Function Code for V·CUN=2

If more than one family is being used in this system, IR+3 will also be the index to the FAM + table to establish the correct validation table. The common deck, COMSACC, is used to search the validation file for the user number, and, if the password given matches the one in the validation file, the user index is placed in the response word (LINF + index + 1). If the user number or password is invalid, the user index is set to 0.

The format of the request word (offset by index) is:

LINF=	Seven Character User Number	0
PWLT=	Password	0
FAMT=	Family Name	

The format of the response word (offset by index) is:

24	24	12
Generated User Code (GUC)	User Index	R

GUC is used by EI200 to uniquely identify terminal entries.

R = 1 user logged in
R = 3 user was already logged in

16.7.3 Make Initial Job File Entry (MJE)

18	6	6	18	12
XSP	CP#	0	Relative Address of INPUT FET	Function Code for V·MJE=3

FET+7 has the job priorities necessary. This routine will enter the job INPUT file (job into input queue) into the system FNT/FST.

If successful, the FNT/FST entry is created, the system sector is written, the FNT address is entered into FET+4, and the status in FET+0 is set to 15 (write complete).

Figure 16-9 is a flowchart of the entry to process the function code. The routines used are:

0AV – Validate user

0BF – Begin file

The local routine, DRJ, is used to drop jobs after DJP finds them.

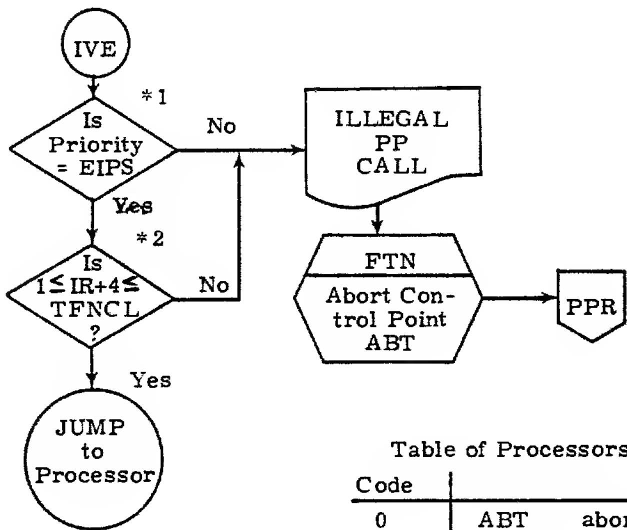


Table of Processors

Code		
0	ABT	abort job
1	DJP	drop job request
2	VUN	log in terminal
3	MJE	job initiation
4	ABT	

*1 get priority of the CP program that called XSP and see if its priority matches the priority of EI200.

*2 is function code legal

Figure 16-9. XSP – Main Entry

16.8 MULTIPLEXER DRIVER (1ED)

The multiplexer driver program is a dedicated PP program designed to drive one 6671 MUX connected with up to sixteen 200 User Terminals or other devices with similar interface characteristics. The designed line rate is 2400 baud.

This program is initially loaded by the EXPORT subsystem executive and is controlled by that executive. The driver will periodically check the system storage move flag and, if necessary, issue a pause function to the monitor. During storage move, no references to central memory are allowed. Activity with the terminals is not disrupted in most cases of storage move because of internal buffering in the driver. If a drop of the EXPORT subsystem is necessary (either because of an operator stop or subsystem malfunction), the executive must set the stop bit in status word zero to cause the I/O driver to release the channel, its reserved equipment, and stop. External to internal codes and vice versa are done via conversion tables.

There are nine major divisions within the driver program.

1. Control driver
   Used to time the I/O cycles to the MUX.

2. Input/output with MUX
   Performs the actual input/output with the MUX when directed by the control driver.

3. Control switch
   Directs the specific activity for each multiplexer port, initiates new activity as directed by the EXEC, and keeps each re-entrant driver active.

4. Poll to connect MUX line
   Probes each active line with all addresses searching for a response. When a response is sensed, the EXEC 1LS is informed.

5. Write message to display
   When directed by the EXEC 1LS, this section is activated to send one message from the MSG buffer to the remote display screen.

6. Print on remote printer
   When directed by the EXEC, this section is activated to transmit one buffer block to the remote printer. The EXEC 1LS is informed at the end of each block so that end of output processing or remote operator directives can be processed if necessary.

7. Read cards from remote card reader
   One block of cards is read from the terminal and the appropriate function is issued to the EXEC 1LS to inform it if more cards, last block, bad codes, etc.

8. Read operator's messages
   This, along with sense terminal condition is used to process input messages from the remote device. The messages are placed in the terminal message buffer for translation by E200CP. Any action required by an operator message is initiated from the EXEC 1LS.

9. Sense terminal condition
   When a connected terminal is otherwise inactive, it is periodically checked for messages originating from the remote terminal or other action required by the remote terminal when not active.

Figure 16-10 shows the 6671 multiplexer port data words. Figure 16-11 is a flowchart of the 1ED main loop.

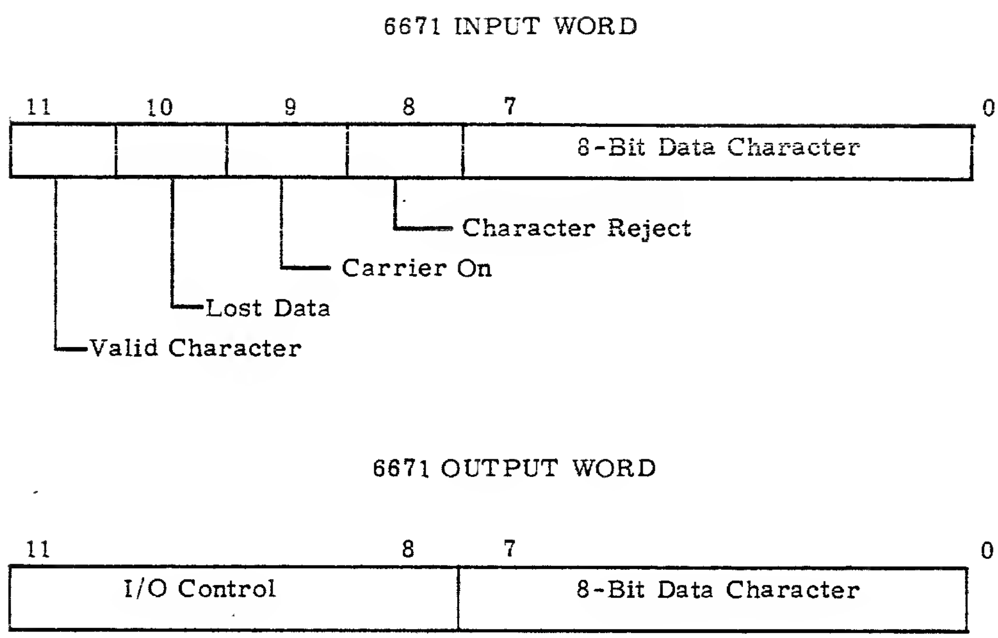
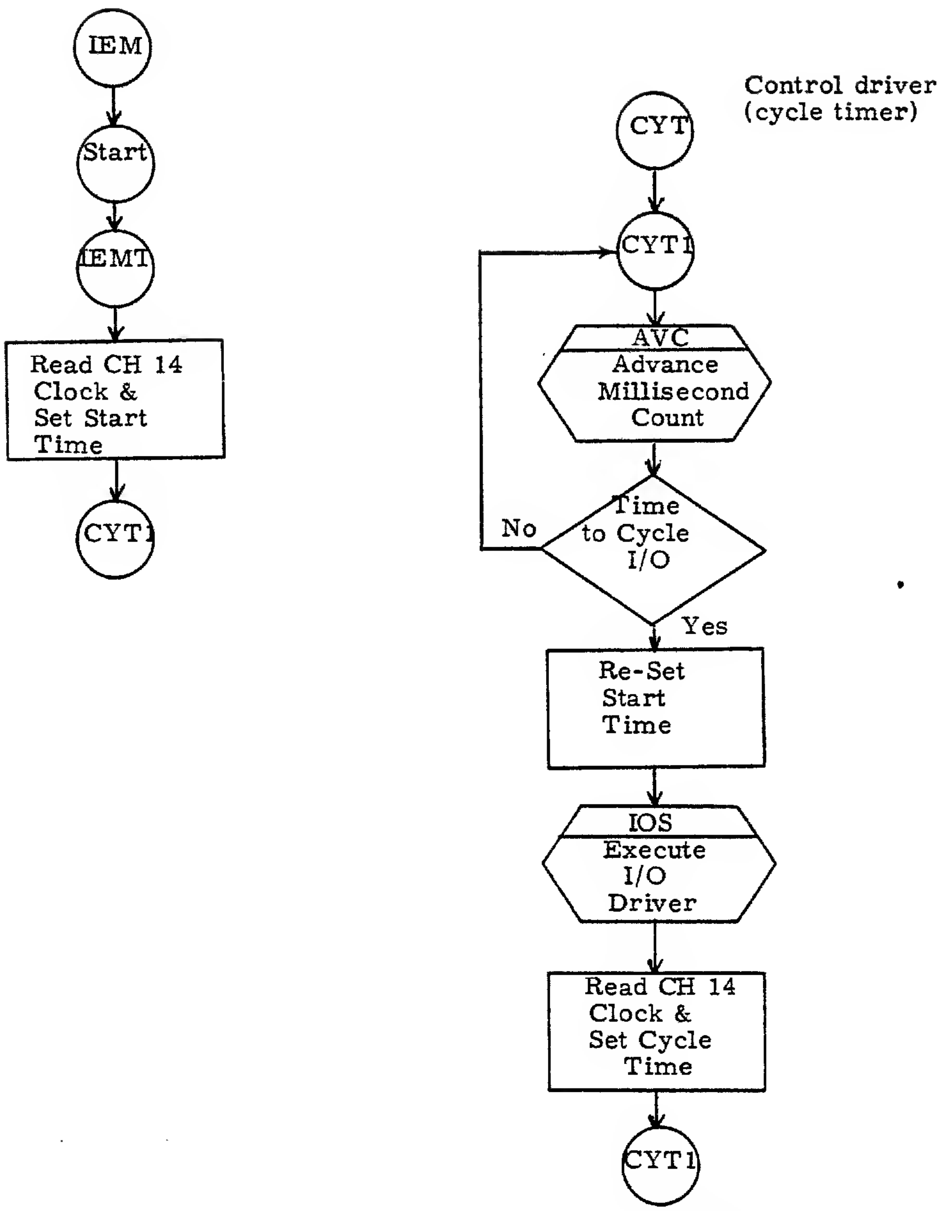
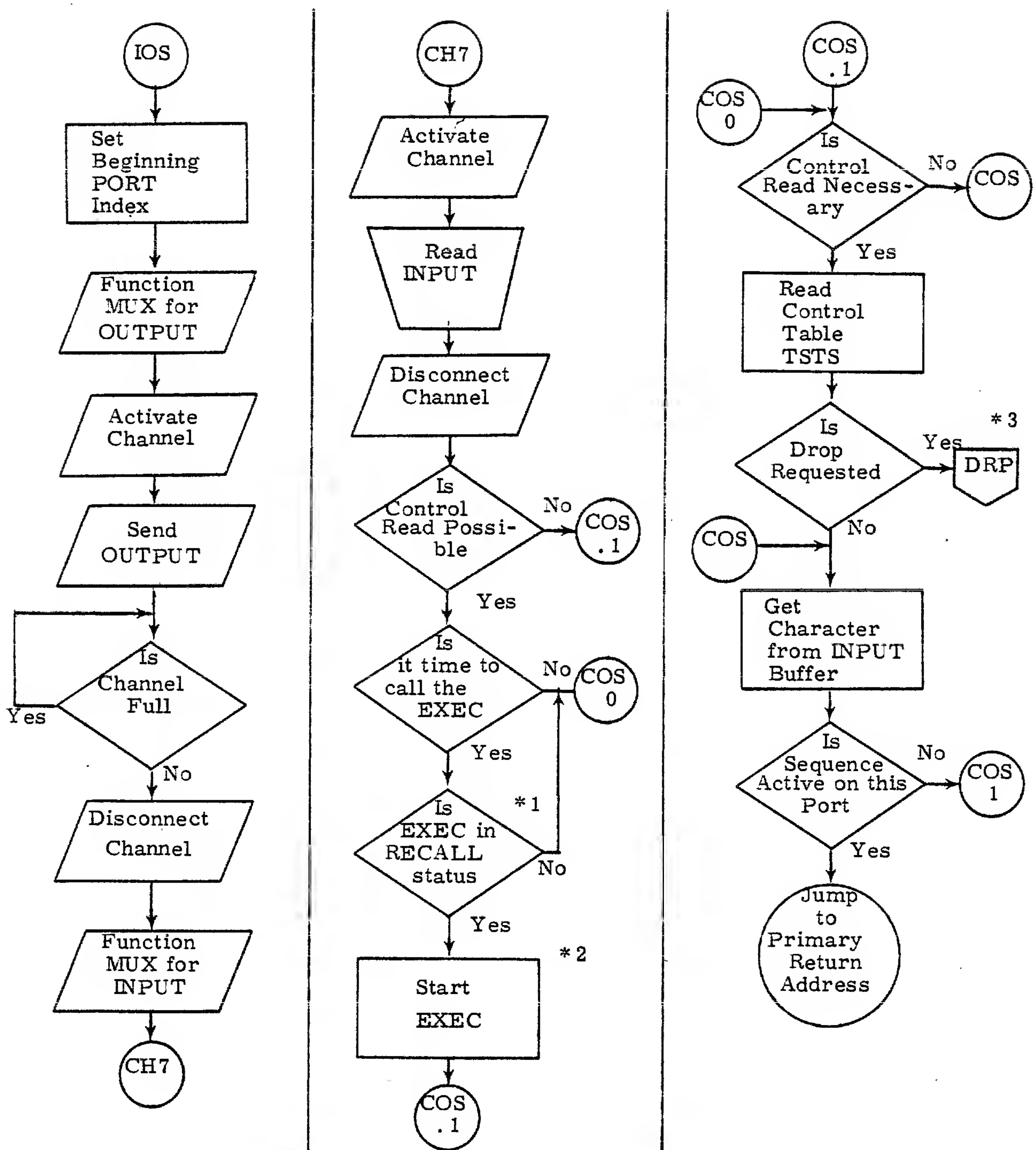


Figure 16-10. 6671 Port Data Word



Initialize routine. Locate a 6671 which is on and free in the EST, insure that it exists and works. If okay, assign channel and modify I/O instructions to use this channel. If not okay, issue error message and drop EI200.

Figure 16-11. IEM – 1ED Main Loop



*1 RA+DRCL word in E200CP. If no input from MUX, E200CP will go into auto call.

*2 Set RA+DRCL word = 1, so E200CP will be taken out of recall.

*3 Drop PP and release MUX channel, and load 1LS to terminate EI200.

Figure 16-11. IEM – 1ED Main Loop (Continued)

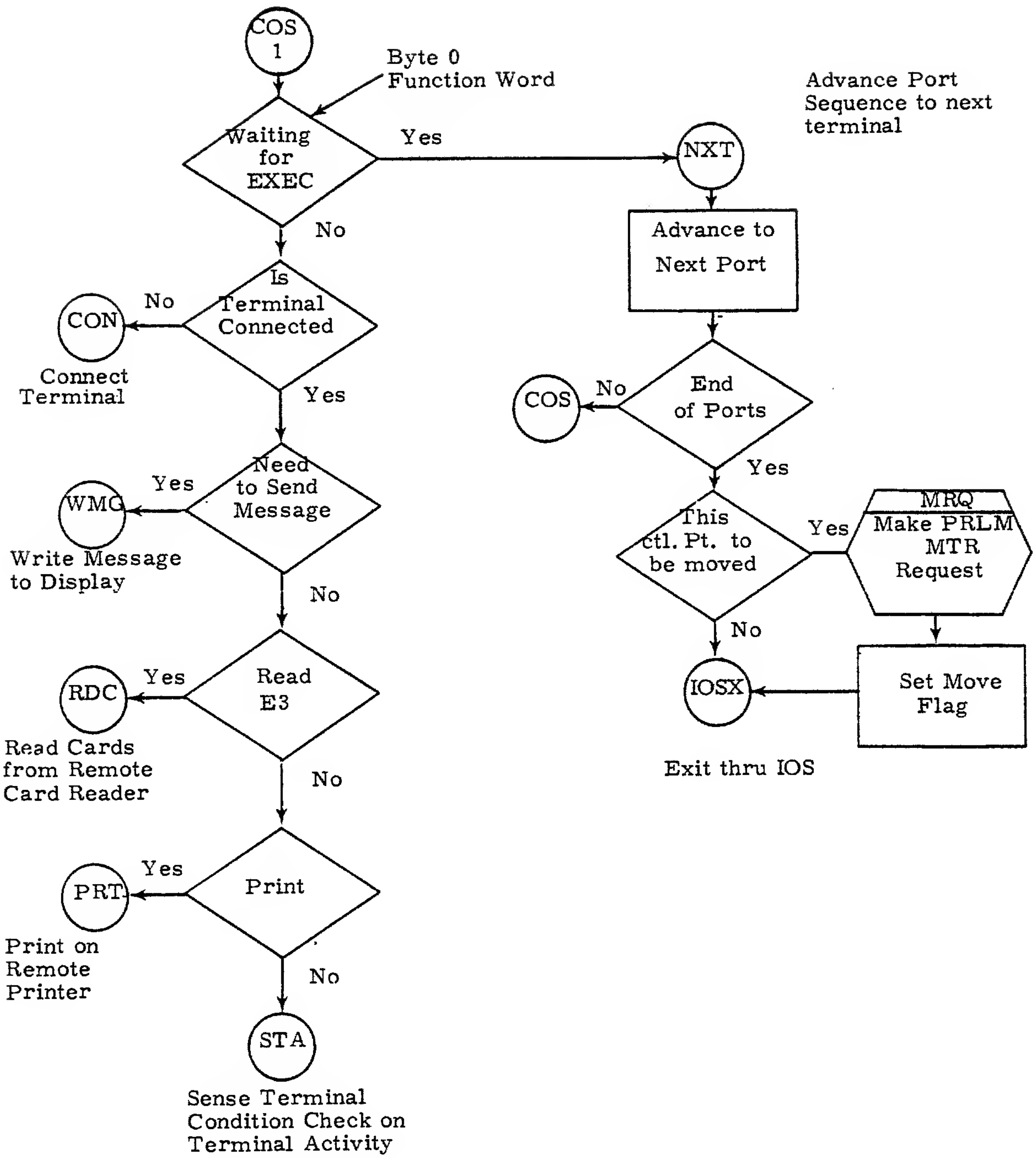


Figure 16-11. IEM – 1ED Main Loop (Continued)

BATCHIO

17.0 INTRODUCTION

The BATCHIO subsystem coordinates communication between the unit record equipment (card reader – CR, card punch – CP, and printer – PR) and the operating system. BATCHIO basically performs the following four functions.

- Reads cards from the card reader, creates the input file, and enters the job into the input queue.
- Locates jobs in the print queue, locates a free printer, and prints the file on this printer.
- Locates jobs in the punch queue, locates a free card punch, and punches the file on this card punch.
- Processes the DSD commands ENDxx, REPEATxx, SUPPRESSxx, RERUN (rerun, nn), on the specified file currently being operated on at the specified buffer point. Buffer point is the term used to associate BATCHIO logical devices with the respective hardware device. Each device is entered into the available equipment table, TAEQ. The index to each entry is the buffer point number. That is, the first entry is buffer point 1, the second entry is buffer point 2, the last device is n. Hence, the DSD command is END1 or END2, etc.

NOTE

BATCHIO (I) display details are explained in Section 4 of the Operator's Guide.

17.1 SUBSYSTEM CONFIGURATION

The subsystem consists of three PP programs and one control point (Figure 17-1).

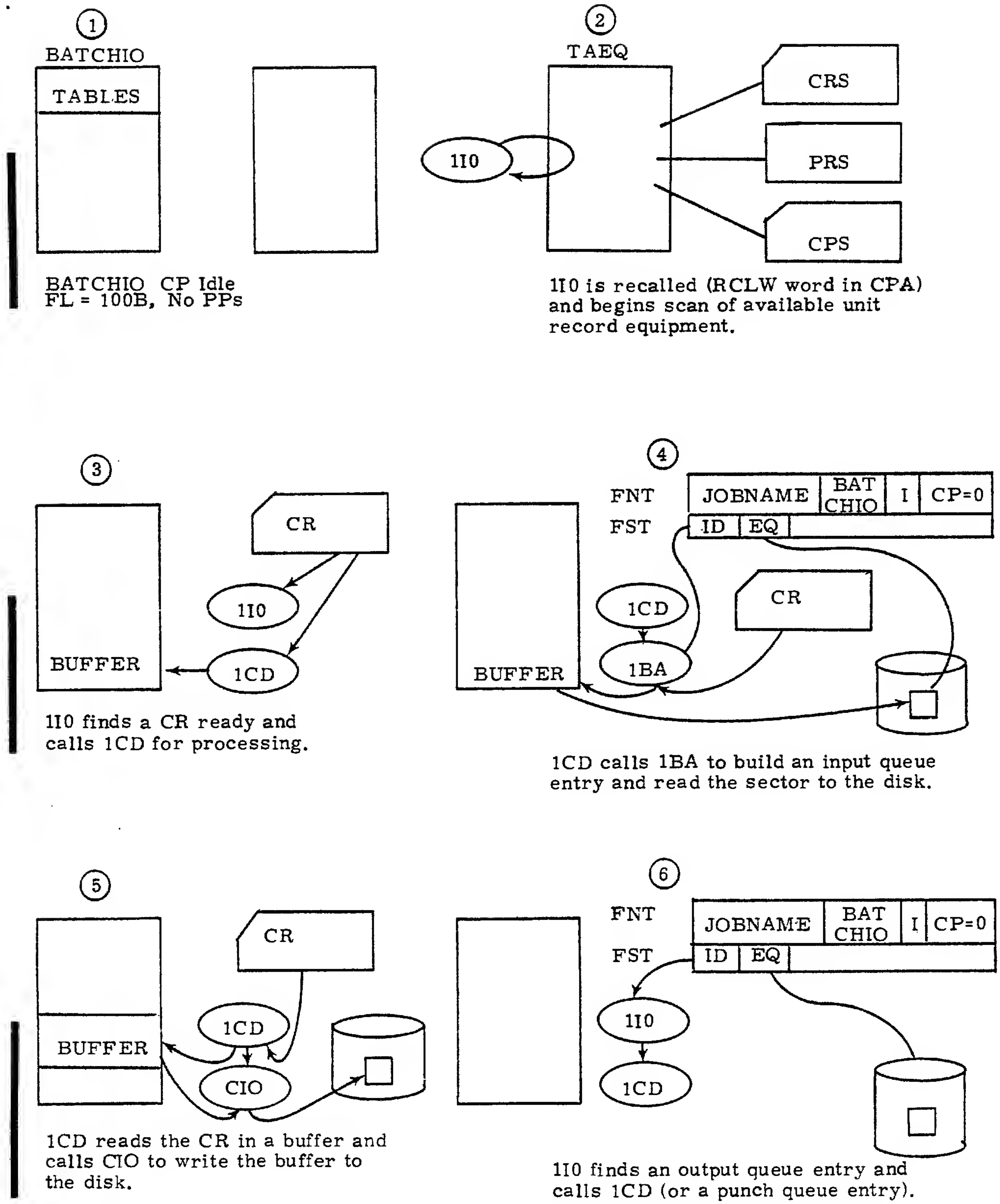
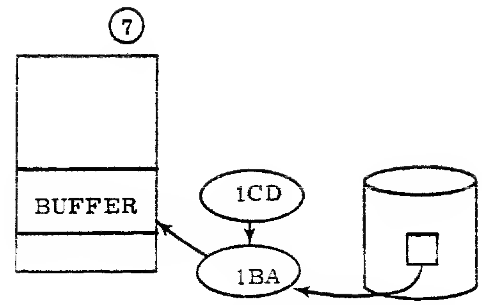
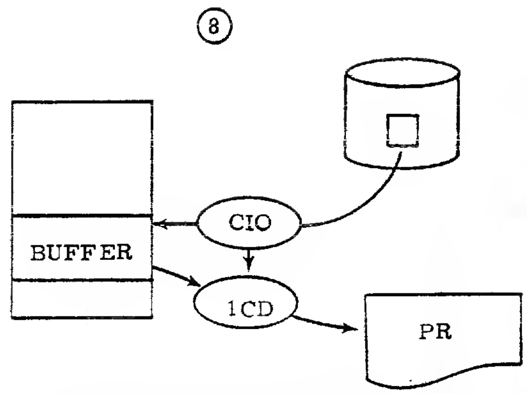


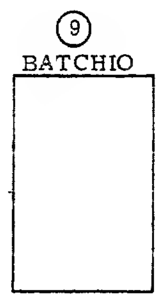
Figure 17-1. BATCHIO Overview



1CD calls 1BA to create a banner page and reads the 1st sector of data from the disk to a buffer.



1CD reads the rest of the file to the CM buffer and prints the data on the printer.



1CD completes and drops, BATCHIO is now idle.

Figure 17-1. BATCHIO Overview (Continued)

17.1.1 Control Point

The control point memory contains no code. The first 100B words are used as a communication area for the three PP routines, and the rest of the central memory (CM) is used for buffer point area, FETs, and buffers. The CM is allocated and deallocated (expanded and contracted) as the need arises. The need arises whenever a device or a BATCHIO type queue (OUTPUT, PUNCH) needs servicing. Each activated device will be assigned one FET and one buffer. Each active device uses a 6-word FET, and each card reader or printer uses a 1002B word buffer, while each card punch uses a 402B word buffer. Figure 17-2 is a diagram of the core layout.

RA + 0	0
RA + 1 RA + 7	DCW – driver control words (1C1)
RA + 10	DRQR – Driver request word
RA + 11 RA + 17	
RA + 20 RA + 27	TEQR – Available equipment table
RA + 30	
RA + (BUFR)	FETs + buffers

Note further that in the idle state BATCHIO has only 100B words of CM, and only the first 30 are of importance. The remaining words are just residue of preceding operations. CM can only be allocated in increments of 100B words.

Figure 17-2. BATCHIO Central Memory Area

17.1.2 BATCHIO Manager

1IO is the BATCHIO CM manager, allocating and deallocating core as devices go active and become idle. In addition, this routine scans the card readers and the OUTPUT and PUNCH queues, and starts up the drivers in 1CD.

17.1.3 Combined Driver

1CD is the combined driver for these three devices. 1CD will call CIO to read and write on mass storage (MS) and 1BA for certain auxiliary functions.

17.1.4 Auxiliary Processor

1BA is the auxiliary processor. It performs processes which would be difficult or impossible to perform in 1CD.

NOTE

All three routines use the common deck COMSBIO for table and communication area specifications (see Section 20).

17.1.5 Central Memory Area

Figure 17-2 illustrates the BATCHIO central memory area. Specific areas of concern are described subsequently.

17.1.5.1 RA + 0 and RA + 1 always remain zero.

17.1.5.2 RA + 1 through RA + 7, DCW are used by 1IO to determine how many copies of 1CD are active and how many requests each one is currently processing. They are allocated backward. That is, the first time 1IO assigns a copy of 1CD, it will set up RA + 1. Then 1IO can assign up to MEQD (currently 10) requests to this 1CD. When this 1CD is working on MEQD requests, it is necessary for 1IO to assign another copy of 1CD. 1IO will therefore set RA + 2, etc.

DCW = 0 means a corresponding copy of 1CD is not active

	18	6	12	24	12
if active, DCW =	1 C D	0	DCW offset	current number of requests	0

17.1.5.3 RA + 10, DRQR is used by 1IO to give 1CD a request.

	12	12	12	24
DRQR =	DCW offset	EST ord.	Buffer point Number	Buffer FWA

DCW offset, RA + DCW offset gives this 1CDs DCW. That is, if DCW offset is 7 (first 1CD called), then RA + 7 is the DCW for this 1CD. If DCW offset is 2 (6th and maximum 1CD call), then RA + 2 is the DCW for this 1CD.

EST ordinal is the ordinal of the device which 1CD must drive.

Buffer number and FWA. The buffers in BATCHIO's field length are allocated in the order desired. If a buffer becomes available, this device will be assigned to this buffer number, otherwise 1IO will allocate more fl and increment the buffer count. The buffer number field is just the sequential number of this buffer. Buffers are threaded via the LIMIT field so that 1CD can count its way to the correct buffer. Buffer FWA is actually the FWA of the FET for this buffer number.

17.1.5.4 RA + 20 through RA + 27 TEQR is an exact copy of the TAEQ table (Figure 17-8) built by 1IO during preset. Every time 1IO is recalled, it restores the TAEQ table (50B PP bytes) by reading the TEQR (10B CM words) into the TAEQ table in its PP memory. (If the status of a unit record equipment is changed (other than OFF when the unit was on) it is not taken into account, and BATCHIO will have to be dropped and recalled to include this equipment.)

17.1.5.5 RA + 30 BUFR points to the first FET of the buffer.

DSD communicates with 1CD for the commands END, REPEAT, SUPPRESS, RERUN, etc. via the control point area (Figure 17-3).

CPA + BFCW points to the first buffer point number.

CPA + BFCWL is the buffer point area.

CPA + BFCW general Format 2 word entry.

							6	6	6
Jobname							repeat count	Code 0,1,2,3,etc	No. of times pru files to SKIP
					S				
6	6	6	6	6	6	6	6	6	6

$$S = \begin{cases} 0 & \text{no suppress} \\ 1 & \text{suppress} \end{cases}$$

DSD COMMANDS for BATCHIO

1 - ENDn.
2 - RERUNn.
none - REPEATn.
3 - SUPPRESSn.
4 - STOPn.
5 - CONTINUEn.
6 - BKSPRUn,y.
7 - BKSPn,y.
10 - BKSPFn,y.
11 - SKIPRUn,y.
12 - SKIPn,y.
13 - SKIPFn,y.

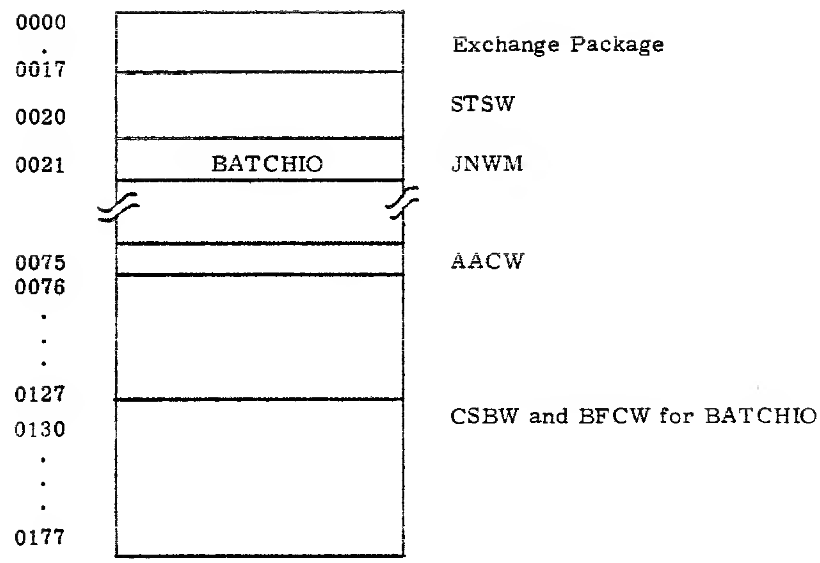


Figure 17-3. Control Point Area

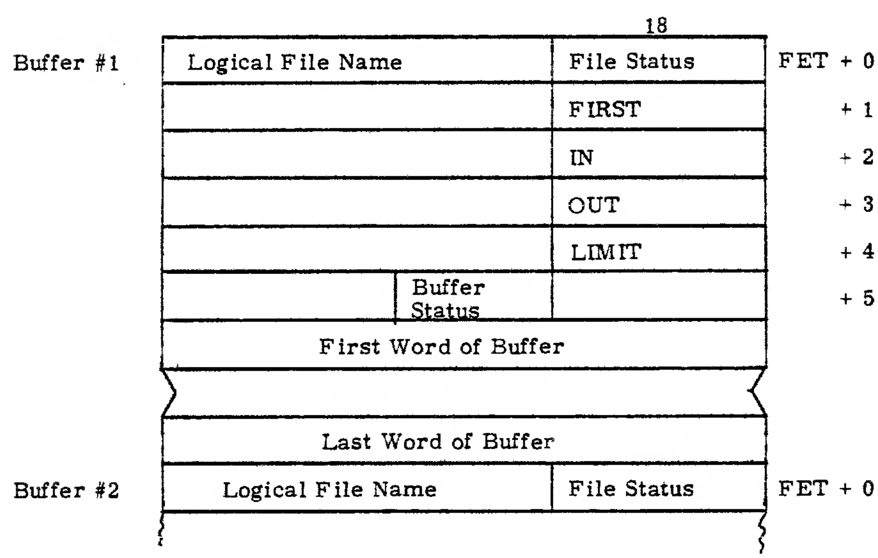


Figure 17-4. Buffers

See Section 7 of the KRONOS 2.1 Reference Manual for more detail. Note that LIMIT = LWA of buffer = FWA-1 of next FET. Hence, when one knows BUFR, and what the buffer number is (not buffer point number) one can easily thread one's way via the LIMIT field to any buffer desired. Also note that FET + 4 and FET + 5 is used by BATCHIO somewhat differently than in a standard FET (that is, 1CD can modify these calls for its own purposes between calls to CIO.

17.2 SUBSYSTEM OPERATION

1IO is a transient PP routine. It recalls itself by always copying its IR into the control point recall register, RLPW, before dropping.

1IO will scan through the TAEQ table processing any device it finds in the ready status. When it has completed one scan, it will recall itself.

When 1IO finds a card reader in ready status, it will initiate a request in DRQR for 1CD to read the card reader. If 1CD must be called, 1IO will recall itself and call 1CD into its PP. If 1CD does not need to be called, it will continue scanning the TAEQ.

When 1IO finds a printer or a card punch in ready status, it will search the FNT for unassigned files in the respective queue (OUTPUT or PUNCH). If the search is fruitless, 1IO will continue its scan of the TAEQ. It is assumed that a job will enter the queues at any time, and this method will assign the output device pointed to by the current TAEQ index. If the search is successful, 1IO will initiate a request for 1CD.

1CD will check the DRQR for a proper request (see 17.4) and, if so, will load the proper driver. 1CD contains drivers for all unit record equipments. It also checks the buffer point word for operator requests. If there is a request for this 1CD for a card reader, the following occurs:

1. 1CD will read one buffer of cards (1000B CM words) and call 1BA to crack the job (1BA calls 2TJ) card (first card in the buffer), set up an FNT/FST entry of type "INPT" (place job in input queue), and via CIO write this buffer onto MS in FET + 5. When complete, 1BA will set the FET status completion bit.
2. 1CD will then read the card reader and transfer the card images to this file created by 1BA via CIO.
3. After the last buffer is transmitted (EOI on card reader) the card reader will be released.

If the request was for a printer, the following occurs:

1. 1CD will call 1BA to create the banner page and place it into the first 20B words of the buffer and indicate completion in the buffer status word, FET + 5.
2. 1CD will then transfer data from the output file into the buffer via CIO and prints the file onto the printer in the proper format.

3. When CIO indicates an EOI status in the buffer, 1CD will call 1BA to place the job accounting information in the buffer.

4. 1CD will complete the printing of this last buffer, release the output file, and release this printer.

5. 1IO may deallocate the buffer if there are no more files to print or get a new job from the output queue and request 1CD again.

If the request was for a card punch, the following occurs:

1. 1CD will call 1BA to create the header card, place it into the buffer, and indicate completion in the buffer status, FET + 5.

2. 1CD will then transfer data from the file into the buffer via CIO, and then punch the cards.

3. When CIO indicates an EOI status in the buffer, 1CD will punch the last set of cards, release the punch file and release the CP.

4. Same as for printer (step 5).

Figure 17-5 is a diagram of the idle state. 1IO will pop in and out of a PP to check the output queue and the status of all card readers. TAEQ, a table of available unit-record equipment accessible by BATCHIO, points to the EST entry of each device.

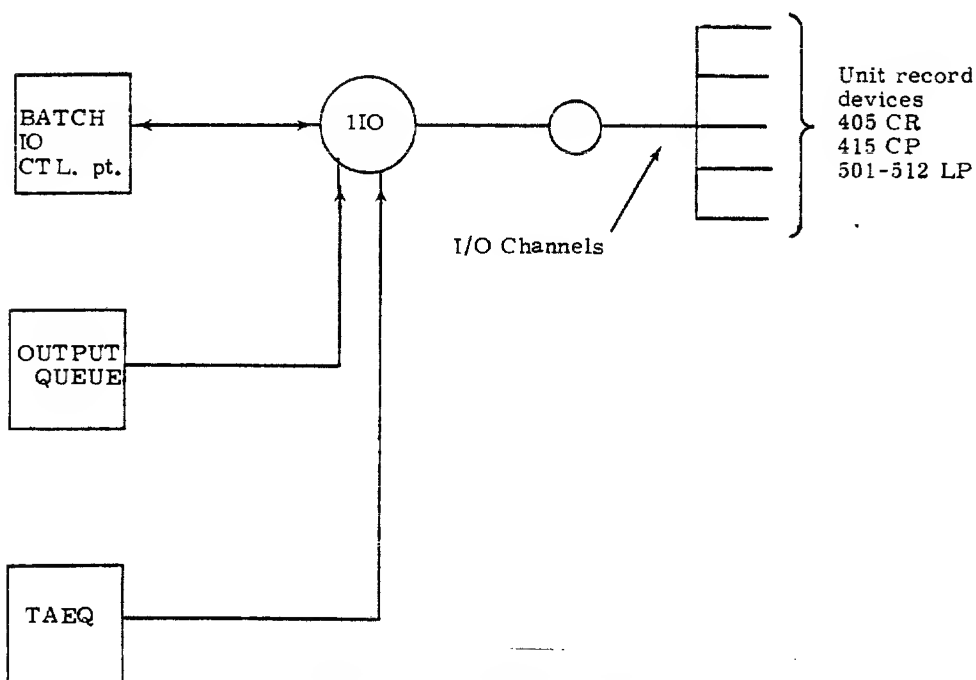


Figure 17-5. BATCHIO Idle State

Figure 17-6 is a diagram of the active state. 1IO reads the TAEQ and the output queues, and builds a request for 1CD in DRQR. 1CD calls CIO and 1BA, which call CIO, etc.

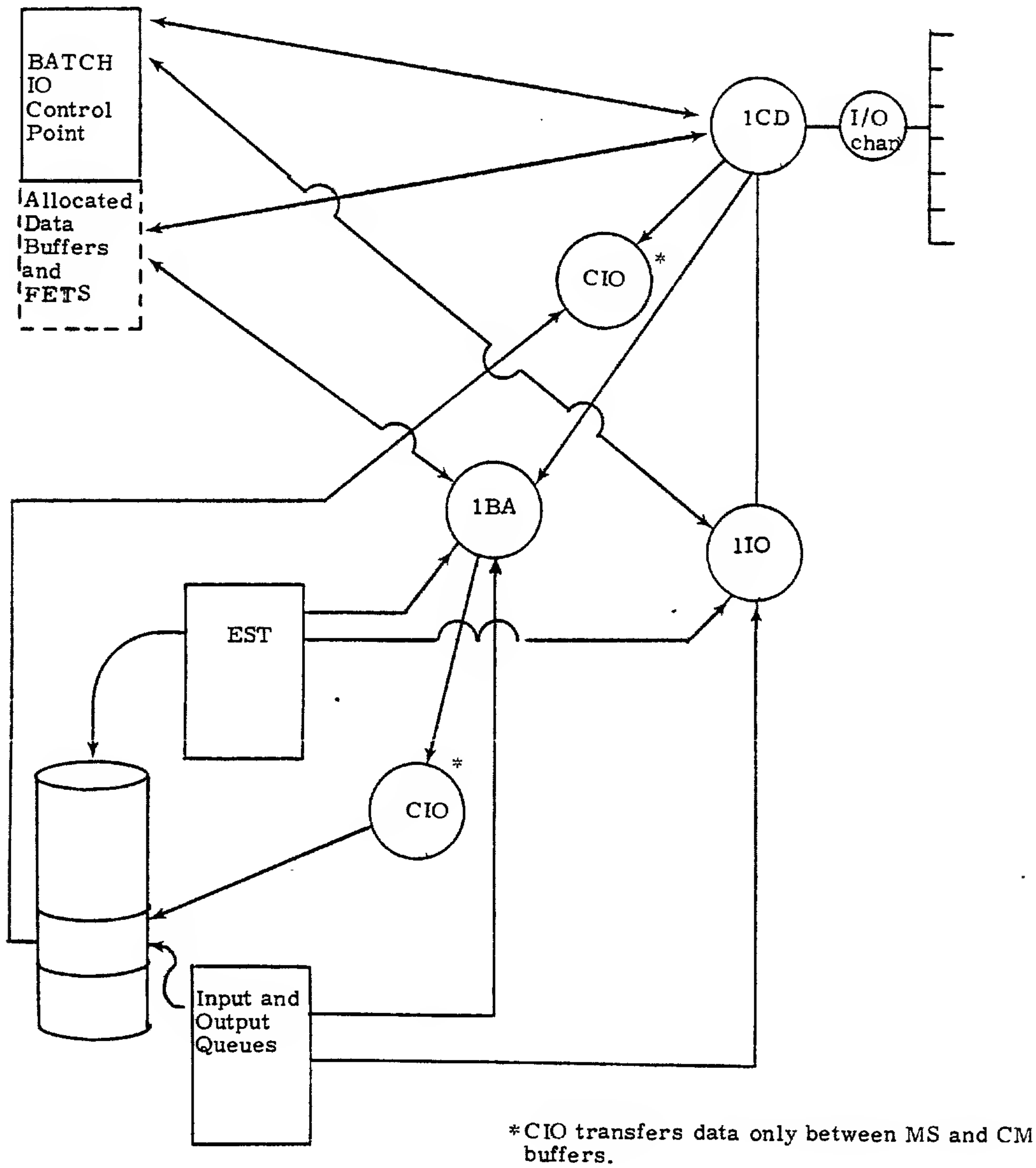


Figure 17-6. BATCHIO Active State

17.3 BATCHIO MANAGER - 1IO

1IO is the executive routine for the BATCHIO subsystem, and performs scheduling of all processes operating at the BATCHIO control point. These include:

1. Searching for the highest priority OUTPUT and PUNCH files.
2. Checking for a ready status on any CRs.
3. Managing of buffer storage and allocating and deallocating CM for the BATCHIO control point.
4. Posting of error condition messages for any of the above.

NOTE

LOAD jobs from tape and DUMP output file to tape is processed by DMQ and LDQ, and not by BATCHIO.

17.3.1 The 1IO call is shown in Figure 17-7.

1R =

1IO	p	cp	a	b	c

where:

p = (0 Preset has not been called (1st time called by 1DS)
(1 Preset has been performed (subsequent calls via the control point recall register RLPW)

cp = Control point number

a, b, c = 0 when first called by 1DS

Figure 17-7. 1TO Call

As 1IO operates, it stores values in these cells (IR+2, 3, 4), and when recalled IR+2, 3, 4 are reset. On recall:

a = IR+2 = scratch direct cell
b = IR+3 = TAEQ index
c = IR+4 = number of buffers allocated = number of requests currently performing.

1IO uses the overlay 3IB to load 3555/512, 595-1 image memory into the 512 if needed.

1IO uses the overlay 3IA for all its subroutines and calls 1CD which is the BATCHIO device driver.

TABLE 17-1. DIRECT LOCATION ASSIGNMENTS

FWA	Code	Value	Location Assignments
20	FS	20 - 24	FST entry (5 locations)
25	BA	25 - 26	Buffer address (2 locations)
30	ES	30 - 34	EST entry (5 locations)
35	ST	35	Equipment status
36	EQ	36	Equipment number
40	CN	40 - 44	CM word buffer (5 locations)
57	FA	57	Address of FST entry
		Assembly Constant	
12	CH	12	Channel number*

*Note that channel 12 appears to be hardwired. This is not the case since 1IO and 1CD will modify all I/O instructions to use the proper channel, via COMPCHI (see Section 20).

When the operator command n. IO, AUTO, or MAINTENANCE is sensed, DSD will call 1DS, which will assign a high number control point (usually n-1) and call 1IO.

1IO will check the p bit in its IR, and since p = 0, it will enter the preset segment.

17.3.2 Preset will perform the following functions.

1. Store the 5 bytes starting at PRSA into the control point area (CPA) JNMW.
2. Store the 5 bytes starting at PRSB into the CPA JCIW.

PRSA = JNMW =	job name = BATCHIO				0
PRSB = JCIW =	CPU priority=1	queue priority ÷ "BIPS"	0	Time Limit 0	0

3. Create the TAEQ available equipment table and copy it to TEQR to enable reloading it for subsequent recalls of 1IO. TAEQ is created by comparing Table 17-2 of equipments, TEQT, to the equipment mnemonics in the EST.

As an equipment is found in the EST which corresponds to any equipment in this table, the TAEQ entry is made. If a device in the EST has an incorrect channel or unit assignment, is turned off, or is rejecting, 1IO will issue one of the following messages:

- EQXX, CHYY, RESERVED. TURNED OFF.
- EQXX, CHYY, NO. 6681. TURNED OFF.
- EQXX. CHYY, REJECT. TURNED OFF.

TABLE 17-2. TEQT TABLE

	12 bits
TEQT + 0	display code LP
+ 1	0
+ 2	display code LQ
+ 3	0
+ 4	display code CP
+ 5	1
+ 6	display code CR
+ 7	2

Each entry in the TAEQ corresponds to the buffer point number. Refer to "I" Display, Section 4 of the KRONOS 2.1 Operator's Guide.

The format for the TAEQ table is shown in Figure 17-8.

	6 bits	6 bits	
TAEQ + 0	Number of Entries in This Table		
+ 1	EST ordinal for 1st eq	eq type	Buffer point 1
+ 3	EST ordinal for 2nd eq	eq type	Buffer point 2
⋮		⋮	
+ 50B	It it exists, EST ordinal for 47B eq	eq type	Buffer point 47B two bytes/entry

EQ TYPE = (0 printer
(1 card punch
(2 card reader

Figure 17-8. TAEQ Table Format

The final function of Preset is to set P=1 in the IR, so that subsequent recalls of 1IO will not cause preset to run, but will cause TAEQ to be loaded from TEQR in the control point memory.

17.3.3 One cycle of 1IO consists of scanning through TAEQ, and either the scan completes or a request to 1CD is made. When one cycle completes, 1IO will recall itself and drop.

When 1IO has a request for 1CD it checks DRQR. If DRQR is nonzero (that is, some copy of 1CD has not yet responded to the last request) 1IO will process error messages, and check central memory allocations. Eventually, DRQR will go to zero. It has been shown that the frequency of requests versus the speed of unit record equipment, versus the speed of 1CD responding, does not necessitate more than a one-word request stack. The time lost while 1IO waits for DRQR to clear is negligible.

When DRQR is zero, 1IO will check the DCW words for an active 1CD. If one is found and byte 2 (number of current requests active) is less than MEQD, 1IO will set up the request in DRQR. Note that up to six copies of 1CD may be active at one time (depending on MXEQ, which is an assembled constant in 1IO and states the maximum number of equipments that can be active at once. Currently MXEQ = 24B so the maximum 1CD copies is three).

If there are no copies of 1CD currently active, or the copies which are active have the maximum current requests MEQD, and this request brings the total current request to less than MXEQ, then 1IO will set up the next DCW word, set up the DRQR word, recall itself, and call 1CD into this PP.

When 1IO is recalled, it will check if the operator desires to drop BATCHIO. If this disable bit has been set (bit 47 in SSTL in CMR), 1IO will not schedule any new 1CD requests. It will wait until all pending requests are complete, process any error messages as they occur, release buffers and all of central memory assigned to the control point, release the control point, and then drop from the PP. If the disable bit is not set, 1IO continues its scan at TAEQ + (IR+3) (Figure 17-9).

Figure 17-10 is a flowchart of the main loop routine and Figure 17-11 is a flowchart of the preset routines.

NOTE

> When 1IO requests 1CD to work on a file in the OUTPUT or PUNCH queues, 1IO will set the CP assigned number field in the FNT to BATCHIO's control point number. This effectively removes the file from the queue so that 1IO will not consider this file again. When 1CD is done and releases the file, the FNT will also be eliminated. If the system crashes and a recovery deadstart is performed, the recovery will attempt to find all files whose FNT queue type is INPUT, OUTPUT, or PUNCH, and will set the CP assigned number back to 0. Therefore, 1IO can find them when BATCHIO is activated.

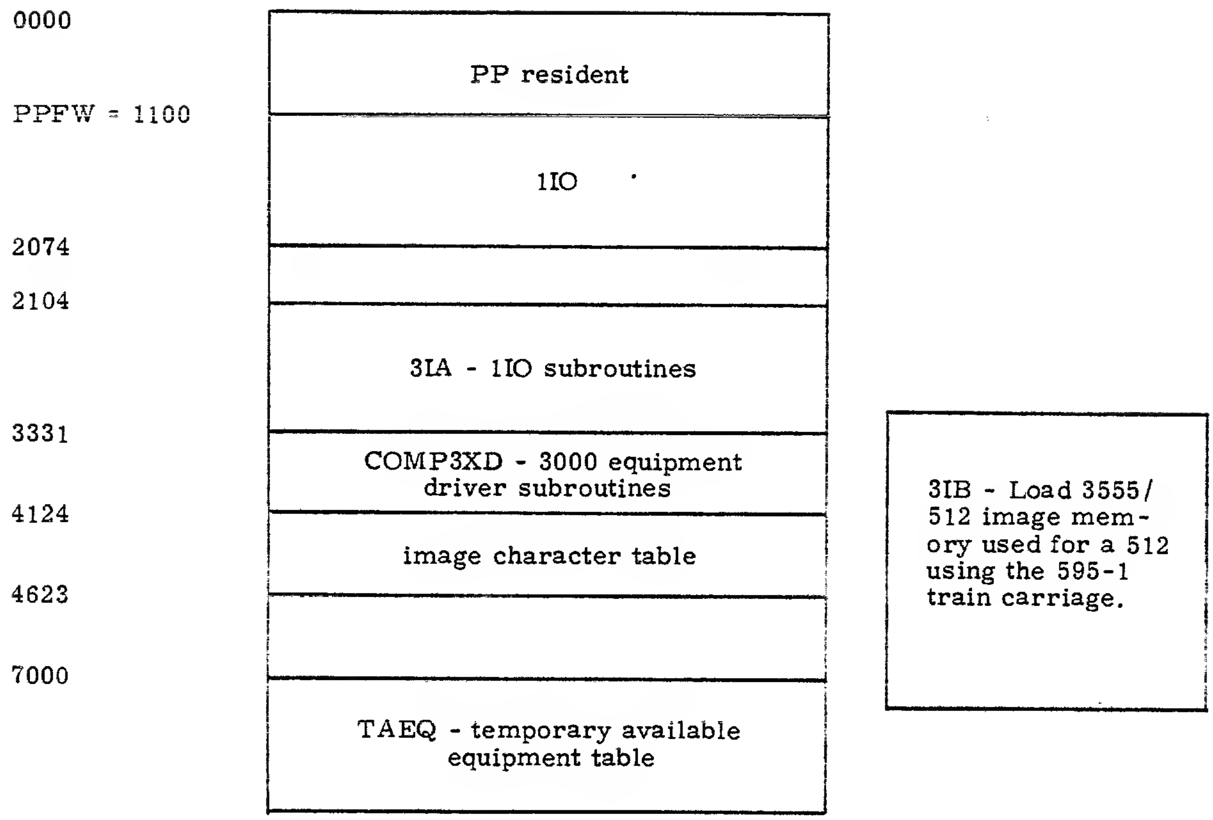
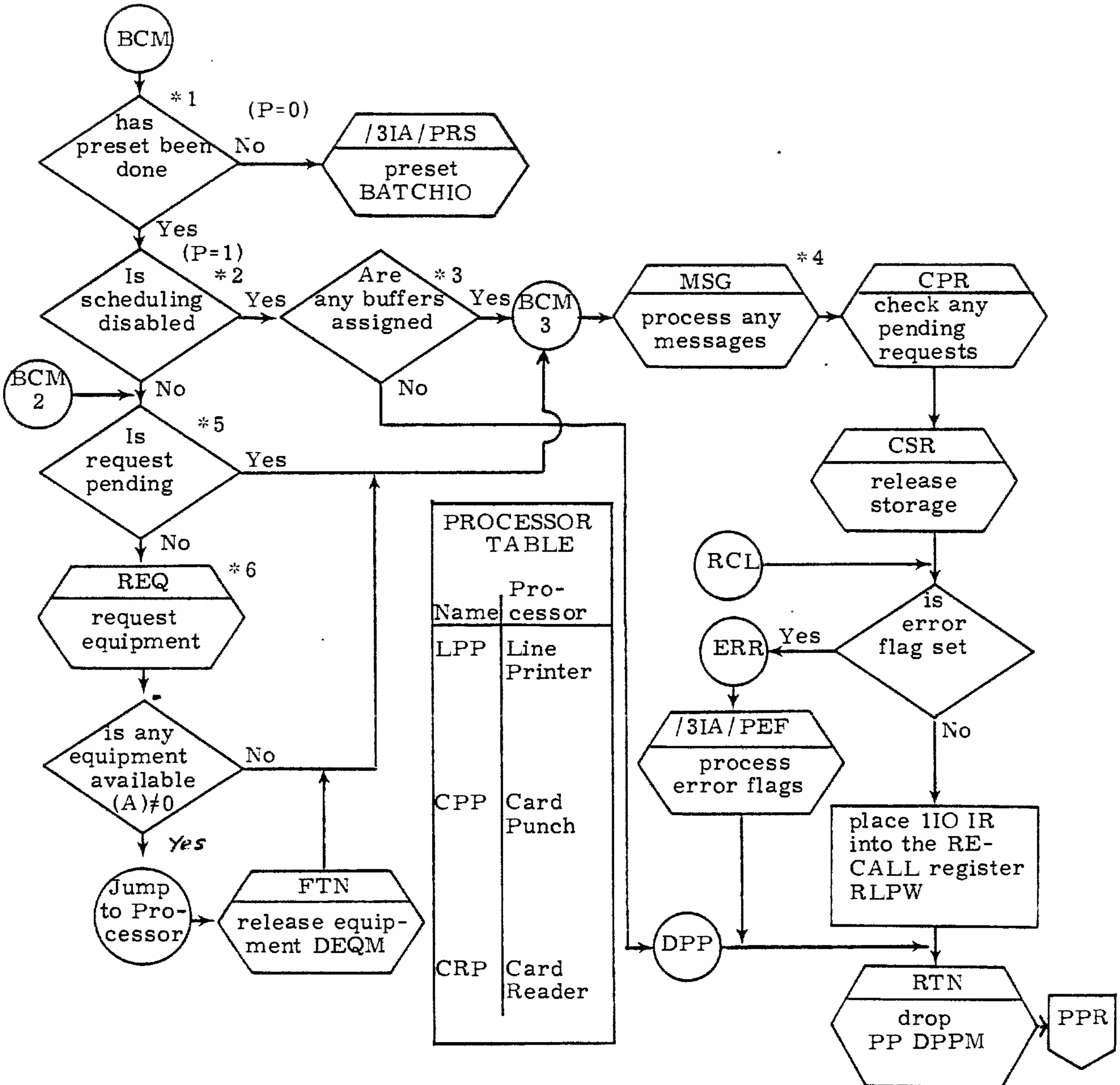


Figure 17-9. 1IO Core Layout



*1 IR+1 = | display code for 1IO | P | CP number | check for P=1

*2 CMR cell SSTL

*3 Is IR+4 negative

*4 Messages are IDLE and xx BUFFERS ACTIVE

*5 Check RA+DRQR

*6 RA+TEQR read EST and check each equipment for status in TEQR table; if some equipment needs to be processed, then (A)≠0, (EQ)=equipment number (ES-ES+4)=EST entry, (IR+3)=equipment index.

Figure 17-10. 1IO - BATCHIO Manager BCM MAIN LOOP

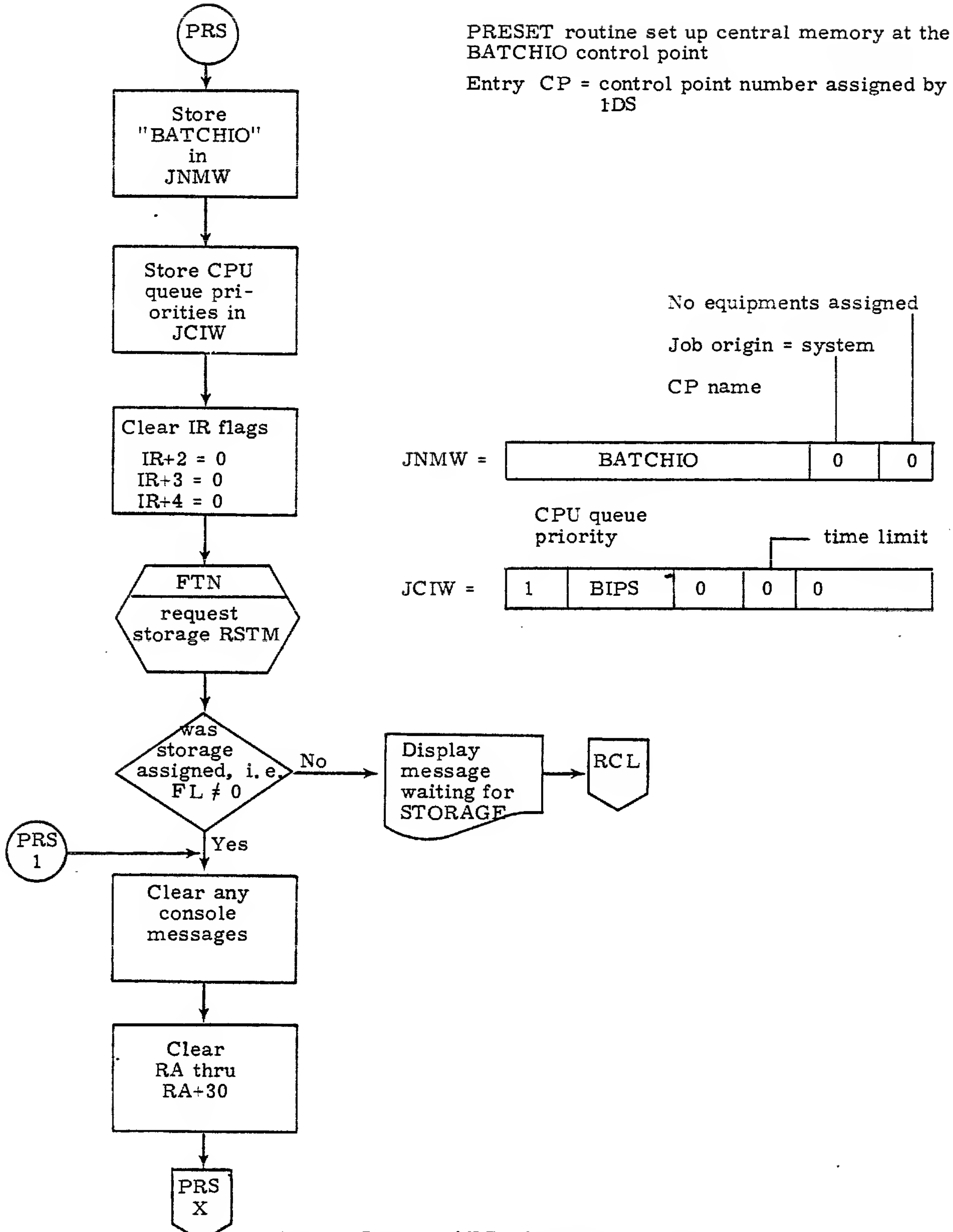


Figure 17-11. BATCHIO Manager PRS for 1IO

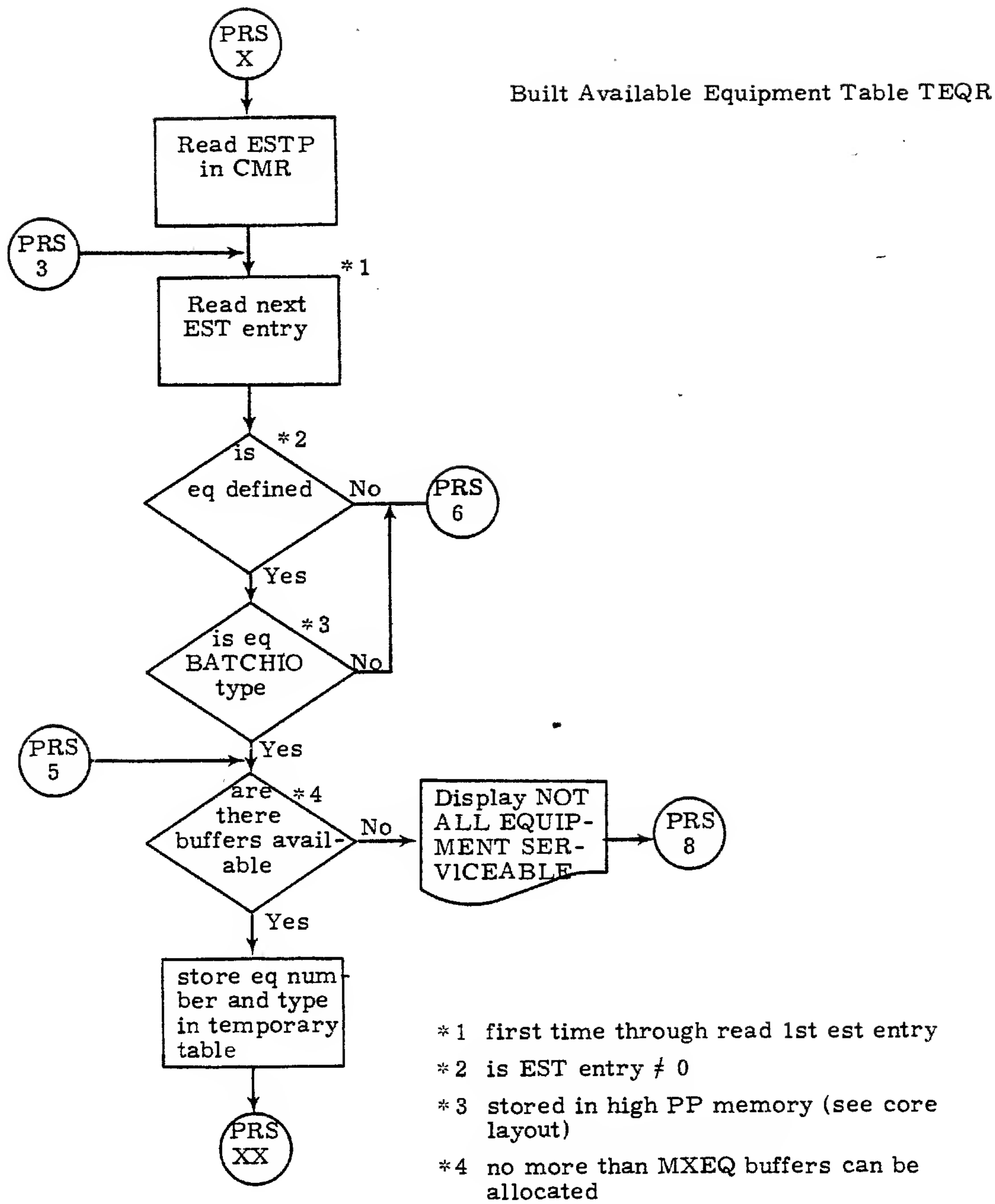
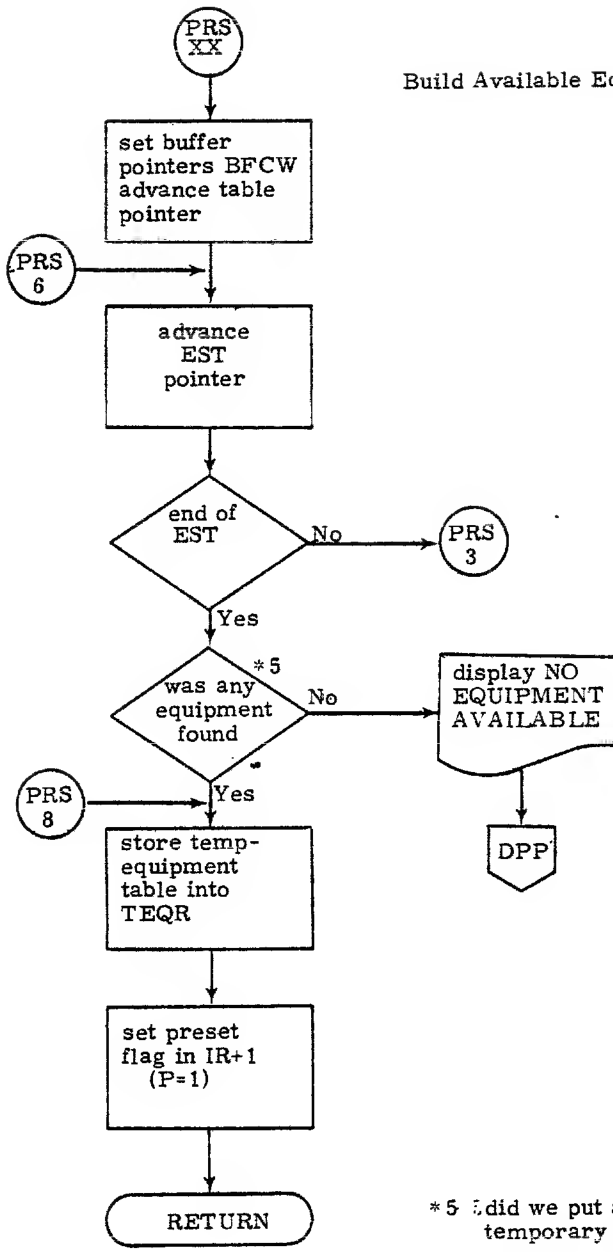


Figure 17-11. BATCHIO Manager PRS for 1IO (Continued)

Build Available Equipment Table - TEQR



*5 did we put any equipment into our temporary table.

Figure 17-11. BATCHIO Manager PRS for 1IO (Continued)

17.4 BATCHIO COMBINED DRIVER – 1CD

The BATCHIO driver, 1CD, can drive up to eight devices of three types (any combination). These three types of devices are:

3256/501/505 - 3555/512	Printer
3446/415	Card Punch
3447/405	Card Reader

Some mass storage transfers and other functions, such as accounting are performed by calling 1BA. Most mass storage transfers are performed by CIO.

A multitude of common decks are used. Among them are:

COMPMAC		
COMSJOT	-	Job output equivalence
COMTDBD	-	Display code to BCD/BCD to Display code
COMTDP9	-	Display code to 029
COMT9DP	-	029 to Display code

All the COMT common decks are simply tables of display code values for each BCD character.

17.4.1 Printer - 3256/501/505 - 3555/512 Driver Characteristics

Line spacing is normally done in the AUTO EJECT mode. That is, creases in the paper are skipped via the 3256 or 3555 automatic line spacing. Thus, it is necessary for AUTO EJECT to be deselected if one wants to use format channels to advance from prior to bottom of form to beyond top of form. An example of this would be with the typical KRONOS format tape which has only one hole in Channel 6 thus providing an eject of up to two pages in order to ensure all banner pages correctly. It should also be noted that deselection of auto eject mode on a 512 will result in deselection of eight lines/inch if previously selected.

The first character of the print line controls the optional formats. This character is not printed. The print line, therefore, consists of up to 136 characters. The first character is not printed if it is recognized as a format control character.

The format control characters and their functions are listed in Table 17-3.

Any format control other than "Q", "R", "S", and "T" are processed once for the line printed.

TABLE 17-3. FORMAT CONTROL CHARACTERS

Char.	Function
C	Skip to format channel 6 after print
D	Skip to format channel 5 after print
E	Skip to format channel 4 after print
F	Skip to format channel 3 after print
G	Skip to format channel 2 after print
H	Skip to format channel 1 after print
Q*	Suppress auto eject
R	Set auto eject
S	Clear 8 lines/inch (512 only)
T	Set 8 lines/inch (512 only)
0	Space 1 line before print
1	Eject page before print
2	Advance to last line of form before print
3	Skip to format channel 6 before print
4	Skip to format channel 5 before print
5	Skip to format channel 4 before print
6	Skip to format channel 3 before print
7	Skip to format channel 2 before print
8	Skip to format channel 1 before print
+	Suppress space before print
-	Space 2 lines before print
/	Suppress space after print
Space	No line control
Other	No line control - character printed

* Q, R, S, and T ignore the remainder of the line.

17.4.2 Card Punch 3445/415 Driver Characteristics

Hollerith cards are punched from a line consisting of up to 140 characters. However, only the first 80 characters of the line are actually punched. The display code to 026 conversion is accomplished by a display code to BCD conversion in the driver followed by the hardware BCD to Hollerith conversion in the 3446. On the other hand, the display code to 029 conversion is accomplished by a display code to binary column images in the driver. These column images are then punched as an absolute binary card.

Binary data are punched in the following format:

Column 1	=	Word count and binary card indicator (79)
Column 2	=	Binary data checksum modulo 4095
Column 3 = 77	=	15 central words of data
Column 78	=	Blank
Columns 79-80	=	24-bit binary card sequence number

Absolute binary data are punched 16 central words/card with no special punches.

End-of-record cards contain a 7/8/9 punch in column 1, and the remainder of the card is blank.

End-of-file cards contain a 6/7/8/9 punch in column 1, and the remainder of the card is blank.

Cards offset are as follows:

- The blank card which precedes the deck.
- All end-of-record cards.
- A card on which a compare error was detected will be offset and also the following card. These two cards will be repunched until no error is detected.

17.4.3 Card Reader 3446/405 Driver Characteristics

Hollerith cards are read with trailing spaces deleted. Up to 80 characters may be transferred to the CM buffer. Hollerith-to-display code translation is accomplished by the Hollerith to BCD conversion hardware in the 3447, followed by a BCD to display conversion in the driver.

Hollerith conversion may be changed by any of the following:

Card	Mode Change Indicator
Job card	"26" or "29" punched in columns 79 & 80
7/8/9 card	"26" or "29" punched in columns 79 & 80
6/7/9 card	"26" or "29" punched in columns 79 & 80
5/7/9 card	no punch in column 2 indicates 026 mode, "9" punch in column 2 indicates 029 mode

A mode change is in effect until changed. Default keypunch mode for a job is defined as an installation parameter.

For the 5/7/9 card, the following are valid conversion mode punches in column 2:

Blank	026
9	029
4/5/6/7/8/9	Literal input Cards are read in binary format with no conversion or checking until a card which is identical in all 80 columns is read.

Binary cards must conform to the above specification for punched binary data. An end-of-record consists of a card with 7/8/9 punches in column 1. And end-of-file consists of a card with 6/7/9 punches in column 1. And end-of-information consists of a card with 6/7/8/9 punches in column 1. The 7/8/9 card and the 6/7/9 card signal input keypunch mode conversion. A "26" punched in columns 79 and 80 of either of these cards (or on a job card) indicates that the following cards are in 026 mode. A "29" in columns 79 and 80 indicate a change to 029 mode. Anything else in columns 79 and 80 will not affect the input mode and will be ignored.

17.4.4 The ICD call is:

	IR+0	IR+1	IR+2	IR+3	IR+4
IR =	1 C D	CP	DCW Offset	0	0

The preset routine moves the COMT deck conversion tables which are assembled at the end of the code into PP resident at location MSD, since 1CD does not use any mass storage drivers. This allows 1CD to use high core for transient tables and data. It presets the control point number so it can call 1BA and specify the control point number in 1BA's IR.

After preset (and as long as 1CD is at this PP), the main loop will check DRQR for nonzero. If it is nonzero, the DRQR DCW offset is compared to this 1CD DCW offset in IR+2. If they do not match or DRQR was zero, 1CD will assess the status of its active requests. As a request goes inactive, 1CD will decrement the active request count in its appropriate DCW word. If the active request count goes to zero, 1CD will clear its respective DCW word and drop.

If the DRQR DCW offset was equal to this 1CD, 1CD will start up the proper driver and increment its active request count in its appropriate DCW word.

17.4.5 After each call to CIO or 1BA, 1CD will jump to its main loop. The drivers can be processing many requests simultaneously. They are honored in sequence by 1CD. The limit of MEQD (currently 10B) is all one copy of 1CD can process; and, still continue to drive each device at top speed.

1CD can issue the following two error messages:

- EQXX, CHYY, FCN ZZ REJECT. = A function reject or transmission parity error was detected.
- EQXX, COMPARE ERROR. = Compare error was detected.

where: EQ = equipment type (CP, CR, LP, or LQ)

XX = EST ordinal

YY = Channel

ZZ = Function code

1CD can issue the following operator messages:

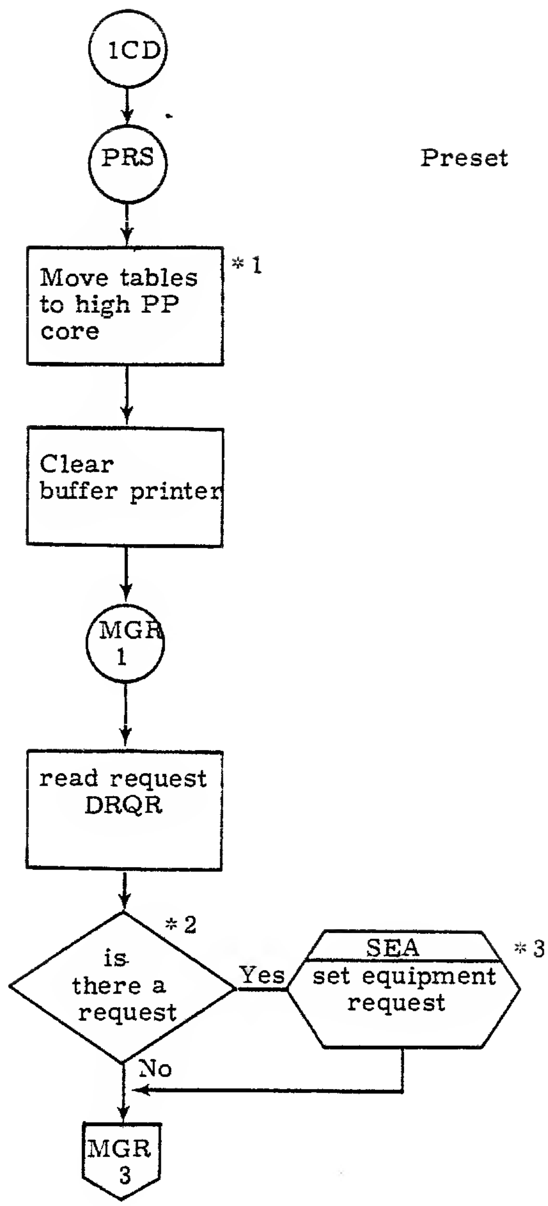
- LPxx. NOT READY. = Printer xx is "not ready."
- LPxx. NO PAPER. = Printer xx has a paper out condition.
- CPxx. NOT READY. = Punch xx is "not ready."
- CRxx. NOT READY. = Reader xx is "not ready."
- CRxx. COMPARE ERROR. - RE-READ 1 CARD. = Reader xx has a card compare error. Operator should re-read the last card in the output stacker.
- CRxx. RE-RD 2 CDS. BINARY ERROR. = Reader xx has encountered a binary card on which the checksum does not check. In order to recover, the operator must re-read the last two cards in the output hopper.

When DSD senses the request ENDxx, REPEATxx, SUPPRESS, or RERUNxx (Rerun job), it calls 1DS which will place the request in the BATCHIO buffer point word (BPW) (If BATCHIO is not active, these commands are ignored) pointed by BFCW in COMTBIO. BFCW points into the BATCHIO control point area which is the control statement buffer.

1CD will periodically check the BPW. If a request is set, it will see if the buffer is busy (that is, data is transferring from buffer to MS or MS to buffer, 1CD must wait for FET status completion bit to be set) and the request is ignored but nor cleared. Eventually, 1CD will find the buffer complete and will process the request as follows:

1. END. Set EOI status in buffer, empty buffer. If printer, 1CD gets the last sector (dayfile) and prints it, then process normal end for equipment.
2. REPEAT. If CR, 1CD ignores the request, else advance repeat count by 1.
3. SUPPRESS. If not PR, 1CD ignores the request, else toggle suppress flag. Yes, this means the operator can unsuppress a listing.
4. RERUN. If CR, 1CD ignores the request, else read FST entry, reset file status, set new priority, and clear control point assignment (i.e., place file back into output queue).

All these functions will generate an appropriate dayfile message. Figure 17-12 is a flowchart showing the main loop of 1CD.



*1 All the COMTXXX tables are loaded at location MSD (600). This loop relocates them directly at the end of the code of 1CD.

*2 Is the (top byte) = (IR+2), i. e., this PP number.

*3 Build table of equipments to check for activity.

Figure 17-12. BATCHIO Driver - 1CD

17.5 BATCHIO AUXILIARY PROCESSOR – 1BA

The auxiliary processor 1BA is called by 1CD to process Mast Storage (MS) transfers and special functions impossible or inconvenient to perform in 1CD. 1BA uses the following routines:

ØBF - Begin file

2TJ - Translate job card

Auxiliary processor 1BA consists of five processors, which are discussed subsequently.

17.5.1 LPR - Load Initial Print Data, Function 1000

This routine (entry LPS) loads the banner page for the job which is being printed. The banner consists of the file name in formatted characters. At exit, the header page and banner page are stored in the buffer starting at FIRST. IN buffer parameter is updated.

17.5.2 LPH - Load Initial Punch Data, Function 2000

This routine (entry PHD) stores an image of the job name in the buffer. This image is in the form of holes which are to be punched on a card to produce a readable deck identification (header card). At exit, the header card consisting of 20B words will be stored in the buffer beginning at FIRST. IN will be updated.

17.5.3 ACT - Process Accounting Information, Function 3000

This routine (entry ACT) will put the accounting information at the end of print buffer. This information is:

"LP XXXXXX.XXXKLN" = Kilo lines printed

"PC XXXXXX.XXXKCD" = Kilo cards punched

"CR XXXXXX.XXXKCD" = Kilo cards read

17.5.4 IIF - Initiate Input File, Function 4000

This routine (entry IIF) will set up the input file and begin to (0BF) build the FNT/FST for it set its type = "INPT", etc. It also calls CIO to write the first buffer. This routine is called to build an input file consisting of the card images read from the card reader. It effectively places the job read from the card reader into the input queue.

17.5.5 BCAX - Subroutine Exit, Function 0 or End-of-Table

This portion of 1BA is the terminus point of the other four routines. Also, if a function is illegal (not in the function table TOPC in 1BA's PP memory), it will fall to this point. BCAX will set the status complete and exit from the PP.

The 1BA call is:

	IR	IR+1	IR+2	IR+3	IR+4
IR =	1 B A	CP number	0	FWA of buffer	

1BA will set the completion bit in the FET (word FET+5) when he is done.

The core layout for the 1BA overlays is shown in Figure 17-13.

1CD will set the function code (1000, 2000, 3000, 4000) in the buffer status word in the FET.

0000	PP resident
PPFW = 1100	1BA main loop
1144	IIF - initiate input file
2000	3BA - ACT - Accounting 3BB - LPD - Printer banner 3BC - PHD - Punch header card

Figure 17-13. 1BA Core Layout

Figure 17-14 is a flowchart showing the main logic for 1BA.

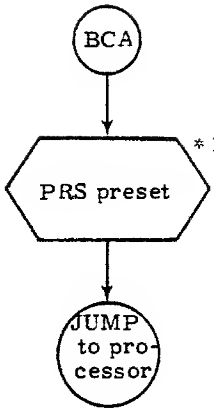
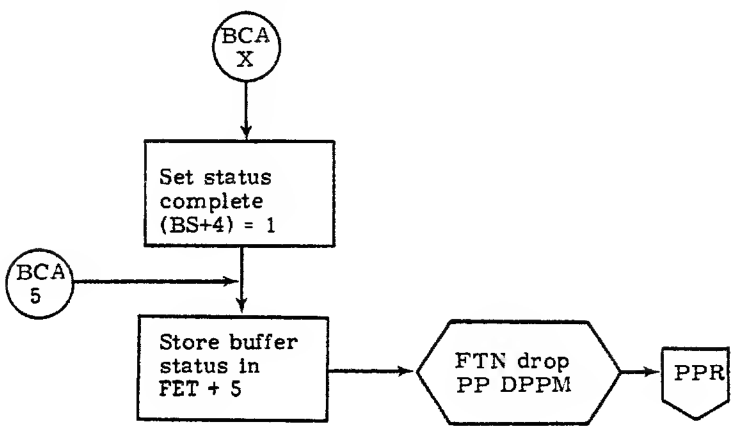


Table of Processors *2

Function	Routine
1000	LPR
2000	LPH
3000	ACT
4000	IIF
0	BCAX

LPH and ACT processors return to BCAX



*1 Set up (FT - LM+1) = Buffer parameters - FET stuf
(FA) = Address of FNT entry
(BS - BS+4) = Buffer status
(A) = (BS+4) = Function request (1000, 2000, 3000, 4000, 0)

*2 Table of processors is TOPC

Figure 17-14. BATCHIO Auxiliary Processor - 1BA Main Entry BCA

The following is a dump of BATCHIO FL and CPA (Level 5):

TABLE 17-4. BATCHIO CPA & FL

Foil	Address		Description
1	2200-2217 2220		Exchange package status word RA = 135500 FL = 2200
	2221 2225	JNMW RLPW	Name = BATCHIO PP Input register, 1TO is in recall.
	2230	MS1W	Line one message is 2 BUFFERS ACTIVE. Note 12 bits of zero indicates end of line
	2267	CSPW	FST Input address in 3313 with EOR flag set. There are no CC so limit = current = 0. The file at 3313 is used by BATCHIO as a dummy input file
	2330	CSBW	This is normally the CC buffer, but since BATCHIO never runs in the CPU it is never advanced. Hence, this is used to control activity at each Buffer Point.
	2330 & 2331		Buffer Point 1 is the CR. Job MORACYI is being read by CD from the CR.
	2332 & 2333		Job DUMPACCXP is being printed by 1CD on an LP.
2	135500		Start of FL for BATCHIO RA+1 is used for scratch, since BATCHIO is never active in the CPU, neither monitor will ever scan its RA+1. 1CD is called by 1IO via an RPPM function. At this time 1CD with offset 1 is running. It has an activity count of 2. As we saw at 2330 and 2332, buffer point 1 and 2 are active.
	135510		The DRQR is empty so 1IO is not communicating to 1CD at this point. In fact, since RPLW is not zero, 1IO is currently in recall.
	135520		TEQR. There are 2 entries LP at est ord 11 and CR at est ord 20.
	1135530		This is the FET for buffer point 1. The fn is DUMPACX.
	135531		FIRST = 36
	135532		IN = 611
	135533		OUT = 777
	135534		LIMIT = 1100 which points us to the next buffer

TABLE 17-4. BATCHIO CPA & FL (Continued)

Foil	Address	Description
2	135535	Buffer status for 1BA
	135536	FWA of buffer. This buffer is full and extends to 136576.
3	136600	FET for buffer point 2. lfn = JMORACY
	136603	LIMIT = 2200 which is the LWA of fl.
	136602 & 136603	IN=OUT=1116=end of useful info in buffer
	136606	FWA of buffer
	136615	LWA of buffer. Note that the card stream is ended with at least 12 bits of zero.
	136616	LWA + 1 of useful data all following is leftover garbage.

ABSOLUTE DUMP FROM 002200 TO 002400 PAGE 1 BATCHIO CPA

002200	00000304001200000000	CD J	00135500000001000001	K A A	00002200000112000343	R AJ C8
002203	00070000000000000000	G	00000000000101000000	AA	00000000000003000343	C C8
002206	00002200000001000052	R A)	00000000000111000343	AI C8	77777777777776000000	!!!!!!-
002211	00000000000000000000		00000000000000000111	A1	22777727700034000000	R!!W+ 1
002214	77777777777777770000	!!!!!!!!	00000000000000000000		05160420000000000000	ENDP
002217	33333337000000000024	0004 T	00010000000013550022	A K R	02012403101117000000	BATCHIO
002222	00017773000000000000	A!>	00003644000000040000	39 D	77777777770000000000	!!!!!
002225	34111751000300000002	1IO(C B	00000000000000000000		00000000000000000000	
002230	55355502250606052223	2 BUFFERS	55010324112605570000	ACTIVE.	01001133140034451405	A I0L 1+LE
002233	00000000000000000000		00000000000000000000		00000000000000000000	
002236	00000000000000000000		00000000000000000000		00000000000000000000	
002241	00000000000000000000		00000000000000000000		00000000000000000000	
002244	00000000000000000000		00000000000000000000		00000000000000000000	
002247	00000000000000000000		00000000000000000000		00000000000000000000	
002252	00000000000000000674	FS	00000000000000000000		00000000000000000000	
002255	00000000000000000000		00000000000000000000		00000000000000000000	
002260	00000000000000000000		00000000000000000000		00000000000000000000	
002263	00000000000000000000		00000000000000000000		00000000000000000000	
002266	00000000000000000000		33134000000000000000	0K5	00000000000000000000	
002271	00000000000000000000		00000000000000000000		00000000000000000000	
002274	77777777777777777777	!!!!!!!!!!	77777777777777777777	!!!!!!!!!!	77777777777777777777	!!!!!!!!!!
002277	00000000000000000000		00000000000000000000		00000000000000000000	
002302	00000000000000000000		00000000000000000000		00000000000000000000	
002305	00000000000000000000		00000000000000000000		00000000000000000000	
002310	00000000000000000000		00000000000000000000		00000000000000000000	
002313	00000000000000000000		00000000000000000000		00000000000000000000	
002316	00000000000000000000		00000000000000000000		00000000000000000000	
002321	00000000000000000000		00000000000000000000		00000000000000000000	
002324	00000000000000000000		00000000000000000000		00000000000000000000	
002327	00000000000000000000		12151722010331110000	JMORACYI	00040000000000000000	D
002332	04251520010330200000	DUMPACXP	00000000000000000000		00000000000000000000	
002335	00000000000000000000		00000000000000000000		00000000000000000000	
002340	00000000000000000000		00000000000000000000		00000000000000000000	
002343	00000000000000000000		00000000000000000000		00000000000000000000	
002346	00000000000000000000		00000000000000000000		00000000000000000000	
002351	00000000000000000000		00000000000000000000		00000000000000000000	
002354	00000000000000000000		00000000000000000000		00000000000000000000	
002357	00000000000000000000		00000000000000000000		00000000000000000000	
002362	00000000000000000000		00000000000000000000		00000000000000000000	
002365	00000000000000000000		00000000000000000000		00000000000000000000	
002370	00000000000000000000		00000000000000000000		00000000000000000000	
002373	00000000000000000000		00000000000000000000		00000000000000000000	
002376	00000000000000000000		00000000000000000000		00000000000000000000	
002401	00140000000000000000	L	00000000000000000000		00070000000000000000	G
002404	00000000000000000000		00000000000000000000		00002400000000000000	T
002407	00000000000000000000		00000000000000000000		00000000000000000000	
002412	00000000000000000000		00000000000000000000		00000000000000000000	
002415	00000000000000000000		00000000000000000000		00000000000000000000	
002420	00000000000014000000	L	00000000000000000000		00000000000000000000	
002423	00000000000000000000		77777777770000000000	!!!!!	00000000000000000000	
002426	00000000000000000000		00000000000000000000		00000000000000000000	
002431	00000000000000000000		00000000000000000000		00000000000000000000	
002434	00000000000000000000		00000000000000000000		00000000000000000000	
002437	00000000000000000000		00000000000000000000		00000000000000000000	
002442	00000000000000000000		00000000000000000000		00000000000000000000	
002445	00000000000000000000		00000000000000000000		00000000000000000000	
002450	00000000000000000000		00000000000000000000		00000000000000000000	
002453	00000000000000000000		00000000000000000000		00000000000000000000	

JNMW

LP

CR

BATCHIO FL

```
ABSOLUTE DUMP FROM 135500      TO      140000      PAGE      1

135500  00000000000000000000              34030411000100020000  1CD1 A B    00000000600000000000
135503  00000000000000000000              00000000000000000000              00000000000000000000
135506  00000000000000000000              00000000000000000000              00000000000000000000
135511  00000000000000000000              00000000000000000000              00000000000000000000
135514  00000000000000000000              00000000000000000000              00000000600000000000
135517  00000000000000000000              00002110220000000000  81BP        00000000600000000000
135522  00000000000000000000              00000000000000000000              00000000000000000000
135525  00000000000000000000              00000000000000000000              00000000000600000600
135530  04251520010330000601  DUMPACX FA  04110000000001000036  DI    A  3  00000000000000000611
135533  00000000000000000777          01  33160000000000001100  ON       I  00024576002400000000  B-- 7
135536  40343640374133343533  5135460120  33333333333333555505  0000000  E  05135460125555555555  EK=EJ
135541  55555555404040403734      555541  36373537403536353534  3424523221  41363634555555554134  6331    61
135544  24523221634100000000  T)ZQ1Y      55555553533555555555    26        40404040404040403733  5555555540
135547  36363636373537353640  3333424235  55555555555540333342        5007  42355555555555404640  72     555
135552  40404036403637404040  5553534555  40404040404040555555  5555555     55535345555555555555    21
135555  55555555374135334040      462055  40404040404040403637  5555555534  37333641555546205555  4036  -P
135560  55555534403600000000    153       55555535345555555555    21        36413642364237343735  3637374142
135563  36413636363736403734  3633343541  55553637374142363334    34467301  35415555555555364136  26     363
135566  36364037353640363737  3354235344  33364136413642555536  0363637  3  33354235344036363755  027215334
135571  55555555364136363641      363336  36403642364136363641  3537363336  36423734555536333635  3741  3032
135574  37363336374100000000  430346      55555535355555555555    22        36403733373437354040  3540414255
135577  40404033334242353637  5550077234  55553540414255555007    2567  /G  72345555555555364136  <1     363
135602  40364036423636373537  5353733424  33363636363637555536  0333334  3  35353733424033333455  224075001
135605  55555555363736413641      343636  37333735364236424040  4042373755  40403440555534363640  5515  1335
135610  42373755551500000000  744  H      55555535365555555555    23        37414236333435413540  4673012625
135613  41423333333333333333  6706000000  55554673012625675555    ->AVUA    55555555555555404040         555
135616  46363335423534403636  5302721533  37363535373342555555  4322407     55302721533432240755   XWQ1IZ70
135621  55555555363633363537      330324  36333637413436364042  3034613357  37374040555533032430  4455  0C7X
135624  34613357445500000000  110.9       55555535375555555555    24        40404040404036373642  5555553437
135627  40404040404040404040  5555555555  55555555553437555555      14      55555555555555403333         500
135632  34404040404040404040  1555555555  40373537353735555550  5424240  /  01555555555542424055  A      775
135635  55555555404040404040      555555  40403636373436403637  5533413534  36373735555555555555  3442
135640  33413534344200000000  062117      55555535405555555555    25        37333637364037333637  4034354034
135643  37343636373536423636  4133423733  55554034354634413342    51251607  37335555555555363736  40     343
135646  37364036413637373436  4353634413  42363636413640555534  7333635  1  34353634413733363555  123164032
135651  55555555364137333735      364042  37333733363737343733  4040344146  37333640555536464240  4035  3575
135654  40344140403500000000  516552      55555535415555555555    26        36373733364037353637  3440354234
135657  36423637364136363640  3734363335  55553440354234373436    15271413  33355555555555373537  02     424
135662  33363636373733364236  0333440373  36404040404242555542  3555577  7  40333440373355557755  501540  !
135665  55555555373337353636      404233  37354040404033413534  4255550621  34423640555540423342  1735  5707
135670  55550621173500000000   FU02       55555535425555555555    27        40404040403435403441  5555512516
135673  33423733344035423437  0740152714  55555555512516074015    (UNG5M    27145555555555403434  WL     511
135676  35363441373336354240  2316403275  33344037334040555551  0154055  (  12316403275015405555  JY#CW/MS
135701  55555555424040344140      755165  40354042334233333333  5257070000  33333333555575516552  0000  2(≠)
135704  57070000000000000000  .G          55555536335555555555    30        34363335404040404040  1302555555
135707  40404040364037353640  5555354235  55551302555555555535    KB     2  42355555555555404040  72     555
135712  40404040403642364036  5555537353  42364137333636555555  7364033     55555537353736403355    424350
135715  55555555404040404040      555555  36403735404040404040  3542555555  40404040555555555535  5555     2
135720  42550000000000000000  7           55555536345555555555    31        36363641373536373641  3336423436
135723  36423734364137343733  3741364140  55553336423436374136    03713463  41405555555555364036  65     353
135726  37364037343637373536  4354134423  42373437353640555535  7414235  2  34354134423741423555  126174672
135731  55555555363636403636      333533  37353637373336363637  4234403334  36373733555533353342  3440  0207
135734  34403334344000000000  150115      34555555555555555555  1           04251520241351241354  DUMP7K(TK=
135737  36333552555555555555  3021        55555555555555555555              55555555555555555555
135742  55555555555555555555              55555555555555555555              04251520241355465526  DUMP7K - V
135745  05225755345555555555  ER. 1       55423750334250353457   74/07/21.  55353357373457363657   20.41.33.
135750  55555520010705555555     PAGE     34375555555555555555  14          00000000000000000000
135753  55550000000000000000              55555536355555555555    32        36373735373336403734  3442403541
```

TEAR
est 11 = CR
est 20 = LP

threaded buffers

FET
LP

FI
BUFFER STATUS

FWA OF BUFFER

ABSOLUTE DUMP FROM 135500 TO 140000 PAGE 4 BATCHIO FL

136512	55555621651500000000	,Q.R	55555534365555555555	13	40404040403541403733	5555526540	
136515	36403633363535414035	3530322652	55555555526540353032	1.52XZ	26525555555555414140	V1 665	
136520	42334041424037363637	7056754334	42343435414040555566	7112655 V	57056754334711265555	.EA=0*IV	
136523	55555555423642344135	737162	33413735363333333333	0642300000	33333333555573716206	0000 >+IF	
136526	42300000000000000000	7X	55555534375555555555	14	34354241404040404040	I276555555	
136531	40404040364036403637	5555353534	55551276555555555535	J- 2	35345555555555404040	21 555	
136534	40404040403734363736	5555541343	42363636403637555555	7333534	55555541343733353455	614021	
136537	55555555404040404040	555556	36403734404040404040	3541555555	40404040555555555535	5555 2	
136542	41550000000000000000	6	55555534405555555555	15	36363640363737333735	3335344042	
136545	37343733373437343642	4140414137	55553335344042414041	02157656	41375555555555373537	64 420	
136550	33363637333636373336	0334033403	37364137333642555542	4364037 7	40334033403436403755	505051354	
136553	55555555364236373641	373436	36423640364237333640	3735374035	36413640555537343637	3635 4134	
136556	35374035363500000000	245232	55555534415555555555	16	37333636364037333636	4033354033	
136561	36363735404040403535	3342555522	55554033354033334255	5025007	55225555555555373436	R 413	
136564	36363736404040404041	3343555556	34373335344240555541	1402175 6	33343555556140217555	012 [5Q2	
136567	55555555364036373734	353441	36413641363736423734	3636343741	36403636555535344136	3533 2163	FET
136572	36343741353300000000	314620	55555534425555555555	17	34333534404241404141	I021576566	CR
136575	37403335403333424040	4502500755	55551021576566450250	HQ.eY*B/	07555555555555334033	0 050	
136600	12151722010331000001	JHORACY A	04110000000001001106	01 A 1F	00000000000000001116	II	
136603	00000000000000001116	IN	33140000000000002200	0L R	00010000020024000001	A B 7 A	BUFFER
136606	12151722560315353333	JHOR,CM200	33335624424242425755	00,TTTTT.	00000000000000000000		STATUS
136611	01030317251624562523	ACCOUNT,US	05225755000000000000	ER.	03172031230206511116	COPYSBF(IN	
136614 FWA	20252456172524202524	PUT,OUTPUT	52550000000000000000	)	00000000000000000000		end of card
136617 OF	05520000000000000000	E) Junk	22052125052324512427	REQUEST(TW	17562623165424271733	0,VSN=TW00	stream
136622 BUF	33345255000000000000	01)	00000000000000000000		00000000000000000000		
136625 FER	00000000000000000000		00000000000000000000		00000000000000000000		
136630	04112333010203000601	OI50ABC FA	04110000000001000036	01 A 3	00000000000000000521	EO	
136633	00000000000000000632	FZ	33220000000000001100	OR 1	00024557006300000000	B+. 3	
136636	01022355555555555555	ABS	55555555553736405555	435	55555555423434355555	7112	
136641	42375033375035415755	74/04/26.	4233503415033415755	70/06/06.	42345033355034375755	71/02/14.	
136644	03172031554241333355	COPY 7600	24012005235524175541	TAPES TO 6	41333355061722150124	600 FORMAT	
136647	57555555555555555555	.	00000000000000000000		55555555555555555555		
136652	55031720314241555555	COPY76	00000000000000000000		55555555555555555555		
136655	55220614545555555555	RFL=	00000000000000000000		55555555373442555555	417	
136660	03201505155555555555	CPMEM	01022355555555555555	ABS	55555555374140355555	4652	
136663	55555555373737335555	4440	42375033375035415755	74/04/26.	42355034355034335755	72/12/10.	
136666	55555555555555555555		03171624221714552017	CONTROL PO	11162455150515172231	IN7. MEMORY	
136671	55252411141124110523	UTILITIES	57555555555555555555	.	00000000000000000000		
136674	55555555555555555555		55041504555555555555	DMO	00000000000000000000		
136677	55555555555555555555		55041520555555555555	DMP	00000000000000000000		
136702	34555555555555555555	1	03012401141707551706	CATALOG OF	55233123240515555555	SYSTEM	
136705	55555555555506111405	FILE	55555555553455555555	1	55555555555555555555		
136710	55555555555555555555		55423750334250353457	74/07/21.	55353357344257403757	20.17.54.	
136713	55555555555520010705	PAGE	55555555553442550000	17	55555555220503555555	REC	
136716	16011505555555555555	NAME	24312005555555555555	TYPE	55551405160724105555	LENGTH	
136721	55555031323251555555	CKSUM	55555504012405555555	DATE	5550317151505162423	COMMENTS	
136724	00000000000000000000		55550000000000000000		55555555555555555555		
136727	55140203555555555555	LBC	00000000000000000000		55555555555555555555		
136732	55141703555555555555	LOC	00000000000000000000		55555555555555555555		
136735	55200203555555555555	PBC	00000000000000000000		55555555555555555555		
136740	55220222555555555555	RBR	00000000000000000000		55555555555555555555		
136743	55270222555555555555	WBR	00000000000000000000		55555555555555555555		
136746	55041520545555555555	DMP=	00000000000000000000		55555555555555555555		
136751	55150614545555555555	MFL=	00000000000000000000		55555555373443555555	418	
136754	04013106111405555555	DAYFILE	01022355555555555555	ABS	55555555541344255555	617	
136757	55555555364041355555	3562	42375033375035415755	74/04/26.	42345033365033345755	71/03/01.	
136762	42365034355034425755	73/12/17.	04251520550401310611	DUMP DAYFI	14052357555555555555	LES.	
136765	00000000000000800000		55555555555555555555		55040604555555555555	DFD	

18.0 INTRODUCTION

In the course of an execution, a PP routine may want to have some special operations performed. Depending on the routines, different areas of core may be available for loading special routines.

In order to have the facility to load a PP routine anywhere in core, the concept of Location Free Subroutines (LFS) is used.

Two macro packages COMPREL and COMPRLI provide the capability for a subroutine to make itself relocating.

18.1 LOCATION-FREE SUBROUTINE-LFS

By convention, any PP routine, whose name begins with a zero is considered a location-free subroutine. A routine which needs an LFS will set LA direct cell 15 to the location where the LFS is to reside, and then calls PLL to load it.

18.1.1 LFS-COMPREL

There are two ways to code an LFS. The first way is by using COMPREL. The user does a *CALL COMPREL, then codes his program. All "M" type instructions will automatically have LA inserted in the d field. Hence, the user may not specify a d field in any "M"-type instruction. In addition, CRM, CWM, AJM, IJM, FJM, EJM, IAM, and OAM cannot be used. If the user wishes to specify an "M"-type instruction without relocation, he must append a "." onto the instruction. Such as "LJM." instead of "LJM". Also, any "M"-type instruction which references a cell defined in SYSTEXT (PPCOM) will not be relocated. If the user wishes to code non-relocatable code after his relocatable code, he merely uses the macro RSTR, which is contained in COMPREL. COMPREL relocates instructions with reference to LA as they are encountered in the code.

18.1.2 LFS-COMPRLI

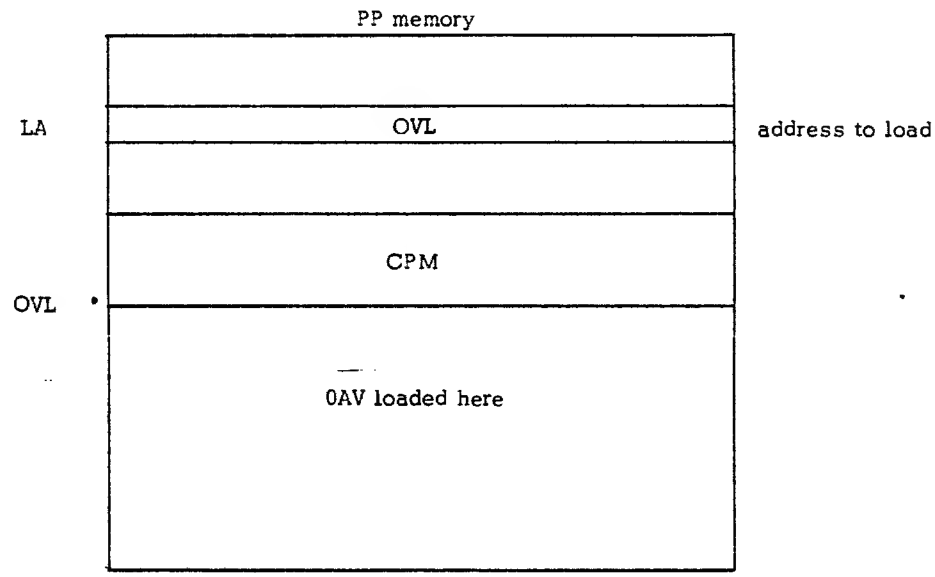
The second method of coding an LFS is to use COMPRLI which relocates indirectly. All the rules of COMPREL are the same with the exception that it is legal to relocate I/O instructions. In addition, the three "C"-type instructions, LDC, ADC, and LMC are also relocatable.

Where COMPREL relocates instructions as it encounters them, COMPRLI builds a remote table using the RMT pseudo op (see COMPASS Reference Manual) containing the address of all instructions that need to be relocated. The first executable statement must be a "RJM. REL, LA". The routine REL is contained in COMPRLI REL will search through the remote table and relocate all instructions whose addresses are stored in the table. Of course, the user must call COMPRLI with an appropriate call statement. (* CALL COMPRLI)

Figures 18-1 through 18-4 are listings of COMPREL, COMPRLI, and the partial listings of 0DF and 0AV which will serve as examples. As a further note, a current listing of COMPREL, and COMPRLI can be obtained by assembling CALLPPU as shown in Section 21.

18.1.3 To Load in LFS

EXECUTE 0AV, OVL



18.1.4 List of current LFSs.

LOCATION FREE ROUTINES
(AS OF LEVEL 5 - AUGUST, 1974)

0AV - Verify user number
0BF - Begin file
0DF - Drop file
0FA - Called by DF to release FA files
0RF - Update RESEXDF and RESEXVF
0PR - Release PF

```
     0                                  CTEXT  COMPREL - LOCATION FREE OVERLAY MACROS.             COMPREL      1
                              *         COMMENT COPYRIGHT CONTROL OATA CORP. 1970.                 COMPREL      2


                              ***       COMPREL - LOCATION FREE OVERLAY MACROS.                    COMPREL      4
                              *         G. R. MANSFIELO.  70/10/04.                                COMPREL      5


                              ***              APPROPRIATE INSTRUCTIONS ARE RE-OEFINEO SUCH THAT   COMPREL      7
                              *         PROPER COOE IS ASSEMBLED FOR LOCATION FREE OVERLAYS.       COMPREL      8
                              *         THE ORIGINAL OEFINITION OF THE INSTRUCTION MAY BE USEO     COMPREL      9
                              *         WHERE APPROPRIATE, BY APPENDING A *.* TO THE OPCOOE.       COMPREL     10
                              *         IF REL$ = 1 THEN THE USE OF SYSTEXT SYMBOLS IN AN -M-      K21001       1
                              *         INSTRUCTION WILL PERMIT THE USE OF THE -O- FIELO.          K21001       2
                              *         IF *REL$* IS NOT DEFINED IN THE PROGRAM OR IS SET " 1,     K21001       3
                              *         THERE IS NO CHANGE IN THE RELOCATION SCHEME.               K21001       4
                              *                                                                    COMPREL     11
                              *         FOLLOWING INSTRUCTIONS USE *LA* FOR RELOCATION.  IF THESE  COMPREL     12
                              *         INSTRUCTIONS ARE USEO WITH A -D- FIELO, IT IS ILLEGAL.     COMPREL     13
                              *                LJM                                                 COMPREL     14
                              *                RJM                                                 COMPREL     15
                              *                LDM                                                 COMPREL     16
                              *                AOM                                                 COMPREL     17
                              *                SBM                                                 COMPREL     18
                              *                LMM                                                 COMPREL     19
                              *                STM                                                 COMPREL     20
                              *                RAM                                                 COMPREL     21
                              *                AOM                                                 COMPREL     22
                              *                SOM                                                 COMPREL     23
                              *                                                                    COMPREL     24
                              *         FOLLOWING INSTRUCTIONS ARE ILLEGAL.                        COMPREL     25
                              *                CRM                                                 COMPREL     26
                              *                CWM                                                 COMPREL     27
                              *                AJM                                                 COMPREL     28
                              *                IJM                                                 COMPREL     29
                              *                FJM                                                 COMPREL     30
                              *                EJM                                                 COMPREL     31
                              *                IAM                                                 COMPREL     32
                              *                OAM                                                 COMPREL     33


                              **        RELM - OEFINE M-TYPE INSTRUCTIONS TO USE *LA* AS O-PART OF  COMPREL    35
                              *         INSTRUCTION.                                               COMPREL     36
                              *                                                                    COMPREL     37
                              *                                                                    COMPREL     38
                              *         RELM   OPC,COOE                                            COMPREL     39
                              *         ENTRY  *OPC* = INSTRUCTION MNEMONIC.                       COMPREL     40
                              *                *COOE* = OPERATION COOE.                            COMPREL     41
                                                                                                   COMPREL     42
                                                                                                   COMPREL     43
                               RFLM     MACRO  OPC,COOE                                            COMPREL     44
```

Figure 18. 1.   COMPREL

```
OOF - OROP FILE.                                         COMPASS 3.73309.    74/03/07. 16.10.33.     PAGE    11
COMPREL - LOCATION FREE OVERLAY MACROS.                  RELM

                            PURGMAC OPC                                                    COMPREL      45
                     OPC.   PPOP   5,CODE                                                  COMPREL      46
                     OPC    MACRO  M,O                                                     COMPREL      47
                            IF     OEF,REL$,4  If REL$ is defined then assemble the next 4  K21001        5
                            IFEQ   REL$,1,3     instructions                                K21001        6
                            IF     SST,//M,2   If SYSTEXT symbol do not relocate            K21001        7
                            OPC.   M,O                                                     K21001        8
                     .2     SKIP                                                           K21001        9
                     .1     IFNE   O,0                                                     COMPREL      48
                            ERR                D-FIELO NOT ALLOWEO.                        COMPREL      49
                     .1     ELSE                                                           COMPREL      50
                            OPC.   M,LA        Relocated by LA                             COMPREL      51
                            ENOIF                                                          COMPREL      52
                     OPC    ENOM                                                           COMPREL      53
                     RELM   ENOM                                                           COMPREL      54
                                                                                           COMPREL      55
     0                      RELM   LJM,0100                                                COMPREL      56
     0                      RELM   RJM,0200                                                COMPREL      57
     0                      RELM   LOM,5000                                                COMPREL      58
     0                      RELM   ADM,5100                                                COMPREL      59
     0                      RELM   SBM,5200                                                COMPREL      60
     0                      RELM   LMM,5300                                                COMPREL      61
     0                      RELM   STM,5400                                                COMPREL      62
     0                      RELM   RAM,5500                                                COMPREL      63
     0                      RELM   AOM,5600                                                COMPREL      64
     0                      RELM   SOM,5700                                                COMPREL      65


                     **     ILLM - OEFINE CERTAIN M-TYPE INSTRUCTIONS TO BE ILLEGAL.       COMPREL      67
                     *                                                                     COMPREL      68
                     *                                                                     COMPREL      69
                     *      ILLM   OPC                                                     COMPREL      70
                     *      ENTRY  *OPC* = INSTRUCTION MNEMONIC.                           COMPREL      71
                     *             *COOE* = OPERATION COOE.                                COMPREL      72
                                                                                           COMPREL      73
                                                                                           COMPREL      74
                     ILLM   MACRO  OPC,COOE                                                COMPREL      75
                            PURGMAC OPC                                                    COMPREL      76
                     OPC.   PPOP   7,COOE      generate table                              COMPREL      77
                     OPC    MACRO  M,O                                                     COMPREL      78
                            ERR                OPERATION NOT ALLOWEO.                      COMPREL      79
                     OPC    ENOM                                                           COMPREL      80
                     ILLM   ENOM                                                           COMPREL      81
                                                                                           COMPREL      82
     0                      ILLM   CRM,6100                                                COMPREL      83
     0                      ILLM   CWM,6300                                                COMPREL      84
     0                      ILLM   AJM,6400                                                COMPREL      85
     0                      ILLM   IJM,6500                                                COMPREL      86
     0                      ILLM   FJM,6600                                                COMPREL      87
     0                      ILLM   EJM,6700                                                COMPREL      88
     0                      ILLM   IAM,7100                                                COMPREL      89
     0                      ILLM   OAM,7300                                                COMPREL      90
```

Figure 18-1. COMPREL (Cont'd)

```
        ***     RSTR - RESTORE ALL REDEFINED INSTRUCTIONS.          COMPREL     92
        *                                                           COMPREL     93
        *                                                           COMPREL     94
        *       RSTR                                                COMPREL     95
        *       ENTRY  NONE.                                        COMPREL     96
                                                                    COMPREL     97
                                                                    COMPREL     98
                PURGMAC RSTR                                        COMPREL     99
         RSTR   MACRO                                               COMPREL    100
         LJM    OPSYN  LJM.                                         COMPREL    101
         RJM    OPSYN  RJM.                                         COMPREL    102
         LDM    OPSYN  LDM.                                         COMPREL    103
         ADM    OPSYN  ADM.                                         COMPREL    104
         SBM    OPSYN  SBM.                                         COMPREL    105
         LMM    OPSYN  LMM.                                         COMPREL    106
         STM    OPSYN  STM.                                         COMPREL    107
         AOM    OPSYN  AOM.                                         COMPREL    108
         RAM    OPSYN  RAM.                                         COMPREL    109
         SOM    OPSYN  SOM.                                         COMPREL    110
         CRM    OPSYN  CRM.                                         COMPREL    111
         CWM    OPSYN  CWM.                                         COMPREL    112
         AJM    OPSYN  AJM.                                         COMPREL    113
         IJM    OPSYN  IJM.                                         COMPREL    114
         FJM    OPSYN  FJM.                                         COMPREL    115
         EJM    OPSYN  EJM.                                         COMPREL    116
         IAM    OPSYN  IAM.                                         COMPREL    117
         OAM    OPSYN  OAM.                                         COMPREL    118
                ENDM                                                COMPREL    119
                ENDX                                                COMPREL    120
```

Return original definitions of these instructions by reseting all of these to normal. Thus all code following RSTR will not be relocated

```
        ****    DIRECT LOCATION ASSIGNMENTS.                        ODF         44
                                                                    ODF         45
                                                                    ODF         46
    57   FA     EQU    57         ADDRESS OF FST ENTRY              ODF         47
        ****                                                        ODF         48
```

Figure 18-1. COMPREL (Cont'd)

```
OAV - VERIFY USER NAME.                               COMPASS 3.73309.     74/03/07. 16.09.44.      PAGE    10
COMPRLI - RELOCATABLE OVERLAY MACROS.

     0                             CTEXT  COMPRLI - RELOCATABLE OVERLAY MACROS.                COMPRLI
                          *        COMMENT COPYRIGHT CONTROL OATA CORP. 1970.                  COMPRLI


                          ***      COMPRLI - RELOCATABLE OVERLAY MACROS.                       COMPRLI
                          *        G. R. MANSFIELO.  70/10/04.                                 COMPRLI


                          ***             APPROPRIATE INSTRUCTIONS ARE RE-OEFINEO SUCH THAT    COMPRLI
                          *        PROPER COOE IS ASSEMBLEO FOR RELOCATABLE OVERLAYS.          COMPRLI
                          *        A RELOCATION TABLE IS GENFRATEO FOR ALL INSTRUCTIONS        COMPRLI
                          *        WHICH MUST BE RELOCATEO.  -M- TYPE INSTRUCTIONS ARE         COMPRLI
                          *        RELOCATEO USING (LA) IF POSSIBLE.                           COMPRLI
                          *        IF THE SYMBOL *REL$* IS SET NON-ZERO, ALL -M- INSTRUCTIONS  COMPRLI
                          *        WILL BE RELOCATEO BY THE RELOCATION TABLE.                  COMPRLI
                          *        THE ORIGINAL OEFINITION OF THE INSTRUCTION MAY BE USEO      COMPRLI
                          *        WHERE APPROPRIATE, BY APPENDING A *.* TO THE OPCOOE.        COMPRLI


                          **       RLIM - OEFINE RELOCATION FOR -M- TYPE INSTRUCTIONS.         COMPRLI
                          *                                                                    COMPRLI
                          *                                                                    COMPRLI
                          *        RLIM   OPC,COOE                                             COMPRLI
                          *        ENTRY  *OPC* = INSTRUCTION MNEMONIC.                        COMPRLI
                          *               *COOE* = OPERATION COOE.                             COMPRLI
                                                                                               COMPRLI
                                                                                               COMPRLI
                          RLIM     MACRO  OPC,COOE                                             COMPRLI
                                   PURGMAC OPC                                                 COMPRLI
                          OPC.     PPOP   5,COOE                                               COMPRLI
                          OPC      MACRO  M,D                                                  COMPRLI
                                   LOCAL  A                                                    COMPRLI
                          .1       MICRO  1,1, M                                               COMPRLI
                                   IFC    GE,$".1"$0$,3                                        COMPRLI
                                   IFC    LE,$".1"$9$,2                                        COMPRLI
                                   OPC.   M,O                                                  COMPRLI
                          .1       SKIP                                                        COMPRLI
                                   IF     OEF,//M,2                                            COMPRLI
                                   OPC.   M,O                                                  COMPRLI
                          .2       SKIP                                                        COMPRLI
                                   IFEQ   REL$,,3                                              COMPRLI
                                   IFEQ   O,,2                                                 COMPRLI
                                   OPC.   M,LA                                                 COMPRLI
                          .3       SKIP                                                        COMPRLI
                                   OPC.   M,O                                                  COMPRLI
                          A        EOU    *-1                                                  COMPRLI
                                   RMT                                                         COMPRLI
                                   USE    REL } Builds a remote table with all locations of all instruc-   COMPRLI
                                   CON    A   } tions of this type to relocate using REL       COMPRLI
                                   RMT                                                         COMPRLI
                                   ENOIF                                                       COMPRLI
```

Figure 18-2. COMPRLI

Also, if -d- field specified, the m field is relocated by LA using the remote table and subroutine REL.

i.e., CRM m, d

so, REL will redefine it during execution as

61d m+(LA) which is CRM m+(LA), I

unless m is a SYSTEXT symbol, then there is no relocation.

If there is no -d- field specified, then just set -d- field = (LA) and don't list in the remote table.

i.e., CRM m will be relocated at assembly time as CRM m, (LA) and REL will not further relocate this instruction during execution.

```
                         OPC      ENOM                                         COMPRLI
                         RLIM     ENOM                                         COMPRLI
                                                                               COMPRLI
   0                              RLIM    LJM,0100                             COMPRLI
   0                              RLIM    RJM,0200                             COMPRLI
   0                              RLIM    LOM,5000                             COMPRLI
   0                              RLIM    AOM,5100                             COMPRLI
   0                              RLIM    SBM,5200                             COMPRLI
   0                              RLIM    LMM,5300                             COMPRLI
   0                              RLIM    STM,5400                             COMPRLI
   0                              RLIM    RAM,5500                             COMPRLI
   0                              RLIM    AOM,5600                             COMPRLI
   0                              RLIM    SOM,5700                             COMPRLI


                         **       RLIO - OEFINE RELOCATION FOR -I/O- INSTRUCTIONS.     COMPRLI
                         *                                                     COMPRLI
                         *                                                     COMPRLI
                         *        RLIO    OPC,CODE                             COMPRLI
                         *        ENTRY   *OPC* = INSTRUCTION MNEMONIC.        COMPRLI
                         *                *CODE* = OPERATION COOE.             COMPRLI
                                                                               COMPRLI
                                                                               COMPRLI
                         RLIO     MACRO   OPC,COOE                             COMPRLI
                                  PURGMAC OPC                                  COMPRLI
                         OPC.     PPOP    7,COOE                               COMPRLI
                         OPC      MACRO   M,O                                  COMPRLI
                                  LOCAL   A                                    COMPRLI
                         .1       MICRO   1,1, M                               COMPRLI
                                  IFC     GE,$".1"$0$,3                        COMPRLI
                                  IFC     LE,$".1"$9$,2                        COMPRLI
                                  OPC.    M,O                                  COMPRLI
                         .2       SKIP                                         COMPRLI
                                  IF      OEF,//M,2                            COMPRLI
                                  OPC.    M,D                                  COMPRLI
                         .3       SKIP                                         COMPRLI
                                  OPC.    M,O                                  COMPRLI
                         A        EQU     *-1                                  COMPRLI
                                  RMT                                          COMPRLI
                                  USE     REL                                  COMPRLI
                                  CON     A                                    COMPRLI
                                  RMT                                          COMPRLI
                                  ENOIF                                        COMPRLI
                         OPC      ENOM                                         COMPRLI
                         RLIO     ENOM                                         COMPRLI
                                                                               COMPRLI
   0                              RLIO    CRM,6100                             COMPRLI
   0                              RLIO    CWM,6300                             COMPRLI
   0                              RLIO    AJM,6400                             COMPRLI
   0                              RLIO    IJM,6500                             COMPRLI
   0                              RLIO    FJM,6600                             COMPRLI
   0                              RLIO    EJM,6700                             COMPRLI
   0                              RLIO    IAM,7100                             COMPRLI
   0                              RLIO    OAM,7300                             COMPRLI
```

Figure 18-2. COMPRLI (Cont'd)

```
OAV - VERIFY USER NAME.                                   COMPASS 3.73309.     74/03/07. 16.09.44.     PAGE    12
COMPRLI - RELOCATABLE OVERLAY MACROS.                     RLIC

                 **        RLIC - DEFINE RELOCATABLE FORM FOR -C- TYPE INSTRUCTIONS.    COMPRLI    103
                 *                                                                      COMPRLI    104
                 *                                                                      COMPRLI    105
                 *         RLIC    OPC                                                  COMPRLI    106
                 *         ENTRY   *OPC* = INSTRUCTION MNEMONIC.                        COMPRLI    107
                                                                                        COMPRLI    108
                                                                                        COMPRLI    109
                  RLIC     MACRO   OPC                                                  COMPRLI    110
                  OPC.     MACRO   C                                                    COMPRLI    111
                           LOCAL   A                                                    COMPRLI    112
                           OPC     C                                                    PRLI1        1
                  A        EQU     *-1                                                  COMPRLI    114
                 Remote -->RMT                                                          COMPRLI    115
                 Code      USE     REL  { Builds remote table so REL relocates all instances of   COMPRLI    116
                 COMPASS   CON     A    { these instructions                            COMPRLI    117
                 pseudo-op RMT                                                          COMPRLI    118
                           ENDM                                                         COMPRLI    119
                                                                                        COMPRLI    120
      0                    RLIC    LOC                                                  COMPRLI    121
      0                    RLIC    AOC                                                  COMPRLI    122
      0                    RLIC    LMC                                                  COMPRLI    123


                 ***       RSTR - RESTORE ALL REDEFINED INSTRUCTIONS.                   COMPRLI    125
                 *                                                                      COMPRLI    126
                 *                                                                      COMPRLI    127
                 *         RSTR                                                         COMPRLI    128
                 *         ENTRY   NONE.                                                COMPRLI    129
                                                                                        COMPRLI    130
                                                                                        COMPRLI    131
                           PURGMAC RSTR                                                 COMPRLI    132
                  RSTR     MACRO                                                        COMPRLI    133
                  LJM      OPSYN   LJM.                                                 COMPRLI    134
                  RJM      OPSYN   RJM.                                                 COMPRLI    135
                           PURGMAC LOC.                                                 PRLI1        2
                           PURGMAC ADC.                                                 PRLI1        3
                           PURGMAC LMC.                                                 PRLI1        4
                  LDM      OPSYN   LDM.                                                 COMPRLI    139
                  ADM      OPSYN   ADM.                                                 COMPRLI    140
                  SBM      OPSYN   SBM.                                                 COMPRLI    141
                  LMM      OPSYN   LMM.                                                 COMPRLI    142
                  STM      OPSYN   STM.                                                 COMPRLI    143
                  AOM      OPSYN   AOM.                                                 COMPRLI    144
                  RAM      OPSYN   RAM.                                                 COMPRLI    145
                  SOM      OPSYN   SOM.                                                 COMPRLI    146
                  CRM      OPSYN   CRM.                                                 COMPRLI    147
                  CWM      OPSYN   CWM.                                                 COMPRLI    148
                  AJM      OPSYN   AJM.                                                 COMPRLI    149
                  IJM      OPSYN   IJM.                                                 COMPRLI    150
                  FJM      OPSYN   FJM.                                                 COMPRLI    151
                  EJM      OPSYN   EJM.                                                 COMPRLI    152
                  IAM      OPSYN   IAM.                                                 COMPRLI    153
                  OAM      OPSYN   OAM.                                                 COMPRLI    154
                           ENDM                                                         COMPRLI    155
```

Figure 18-2. COMPRLI (Cont'd)

```
DAV - VERIFY USER NAME.                                   COMPASS 3.73309.    74/03/07. 16.09.44.    PAGE    13
COMPRLI - RELOCATABLE OVERLAY MACROS.                     REL

                 **      REL - RELOCATE INSTRUCTIONS.                                        COMPRLI   157
                                                                                             COMPRLI   158
                                                                                             COMPRLI   159
 5                       ORG     5                                                           COMPRLI   160
              0   REL$   SET     0                                                           COMPRLI   161
                         QUAL    REL                                                         COMPRLI   162
                         USE     REL                                                         COMPRLI   163
                         USE     REL.                                                        COMPRLI   164
                         USE                                                                 COMPRLI   165
                         RMT                                                                 COMPRLI   166
                         USE     REL                                                         COMPRLI   167
                  REL    SUBR                 ENTRY/EXIT                                     COMPRLI   168
                         LDC     .REL-1       SET FIRST ENTRY                                COMPRLI   169
                         ADD     LA                                                          COMPRLI   170
                         STD     T1                                                          COMPRLI   171
                  REL1   AOD     T1                                                          COMPRLI   172
                         LDI     T1                                                          COMPRLI   173
                         ZJN     RELX         RETURN IF END OF TABLE                         COMPRLI   174
                         ADD     LA                                                          COMPRLI   175
                         STD     T2                                                          COMPRLI   176
                         LDD     LA                                                          COMPRLI   177
                         RAI     T2                                                          COMPRLI   178
                         UJN     REL1         LOOP                                           COMPRLI   179
                  .REL   BSS     0                                                           COMPRLI   180
                         USE     REL.                                                        COMPRLI   181
                         CON     0                                                           COMPRLI   182
                         RMT                                                                 COMPRLI   183
                         ENDIF                                                               COMPRLI   184


                         ENDX                                                                COMPRLI   186
```

Load this Code at end of program

Search table of locations of those instructions to relocate. Relocate the instructions pointed at by this table according to its use. Note RLIM, RLIO, and RLIC. Store these addresses in the order in which they are encountered, and no distinction is made as to which MACRO placed the address into the table. Effectively, every instruction pointed to by this list will have the m part relocated by (LA).

Figure 18-2. COMPRLI (Cont'd)

```
DDF - DROP FILE.                                   COMPASS 3.73309.    74/03/07. 16.10.33.    PAGE    2

                          IDENT  DDF,DRFX
                          PERIPH J
           D_M            BASE   MIXED
                          SST
                   COMMENT 73/06/12. 73/10/12. DROP FILE.
                          LIST   X
                          COMMENT COPYRIGHT CONTROL DATA CORP. 1970.


               ***        DDF - DROP FILE.
               *          G. R. MANSFIELD.  70/07/30.
               *          R. E. TATE.       70/11/30.
               *          M. E. MADDEN.     73/04/01.


               ***               DDF IS A LOCATION FREE ROUTINE TO BE USED FOR
               *          DROPPING ANY SYSTEM FILE.
               *
               *          IF FILE IS TYPE *COMMON*, FILE WILL BE RELEASED FOR USE
               *          BY OTHER JOBS.  ALL OTHER FILES WILL BE REMOVED FROM FNT
               *          AND THE ASSIGNED EQUIPMENT OR TRACKS WILL BE RELEASED.
```

CTEXT COMPMAC – PP SYSTEM MACROS.
<u>CTEXT COMPREL</u> – <u>LOCATION FREE OVERPLAY MACROS.</u>

REL$ not defined indicates full relocation

i.e. SYSTEXT symbols are redefined so user must append ".", if they are not to be relocated. (see address 45D)

Figure 18-3. DDF – Example of COMPREL

```
                                 **        DRF - MAIN ROUTINE.                                        ODF         51
                                                                                                      ODF         52
                                                                                                      ODF         53
    5                                      ORG     5         This allows the 5 word header to start at (LA)   ODF   54
    5        0100 0005            DRF      SUBR              ENTRY/EXIT                               ODF         55
    7        3057                          LDD     FA                                                 ODF         56
   10        0474                          ZJN     DRFX      IF NO FST ADDRESS                        ODF         57
   11        6003                 DRF1     CRD     T3        READ FST ENTRY                           ODF         58
   12        1701                          SBN     1         READ FNT ENTRY                           ODF         59
   13        6010                          CRD     CH                                                 ODF         60
   14        3014                          LDD     CH+4      CHECK FILE TYPE                          ODF         61
   15        1071                          SHN     -6                                                 ODF         62
   16        3401                          STD     T1                                                 ODF         63
   17        1717                          SBN     MXFT                                               ODF         64
   20        0614                          PJN     DRF4      IF UNRECOGNIZED                          ODF         65
   21        2000 0133            DRF2     LDC     TFTY      SET FILE TYPE ADDRESS                    ODF         66
   23        3115                          ADD     LA                                                 ODF         67
   24        3501                          RAD     T1                                                 ODF         68
   25        4001                          LDI     T1        PROCESS FILE TYPE                        ODF         69
   26        0406                          ZJN     DRF4      IF PROCESSOR UNDEFINED                   ODF         70
                                                                                                      ODF         71
                                 *         DRF3 IS RE-ENTERED FROM RFA FOR TAPE FILES.                ODF         72
                                                                                                      ODF         73
   27        5415 0032            DRF3     STM     DRFA                                               ODF         74
   31        0215 0031                     RJM     *                                                  ODF         75
                           32     DRFA     EQU     *-1                                                ODF         76
   33        0351                          UJN     DRFX      RETURN                                   ODF         77
   34        1476                 DRF4     MONITOR MXFM      HANG PP                                  ODF         78
   37        0345                          UJN     DRFX                                               ODF         79


                                 **        RTU - RETURN TAPE UNIT.                                    ODF         81
                                 *                                                                    ODF         82
                                 *         ENTRY FOR TAPE FILES -                                     ODF         83
                                 *                 (T4) = UDT ADDRESS                                 ODF         84
                                 *                 (T5 - T6) = TAPE TYPE AND VSN RANDOM INDEX         ODF         85
                                 *                                                                    ODF         86
                                 *         ENTRY FOR NON-FAST ATTACH PERMANENT FILES -                KRA002       2
                                 *                 (T4) = FIRST TRACK                                 ODF         88
                                 *                 (T5) = FILE MODE AND WRITTEN STATUS                ODF         89
                                 *                 (T6) = EQUIPMENT                                   ODF         90
                                 *                                                                    ODF         91
                                 *         ENTRY FOR FAST ATTACH PERMANENT FILES -                    KRA002       3
                                 *                 (T4) = MEANINGLESS                                 KRA002       4
                                 *                 (T5) = FILE MODE AND WRITTEN STATUS                KRA002       5
                                 *                 (T6) = FNT ADDRESS OF FAST ATTACH FILE             KRA002       6
                                 *                                                                    KRA002       7
                                 *         CALLS   *OFA*, *ORF*, OR *ORP*.                            KRA002       8
                                                                                                      ODF         93
                                                                                                      ODF         94
   40        0100 0040            RTU      SUBR              ENTRY/EXIT                               ODF         95
   42        3004                          LDD     T4                                                 ODF         96
   43        5415 0112                     STM     OVLA+2                                             ODF         97
   45        3006                          LDD     T6                                                 ODF         98
   46        5415 0111                     STM     OVLA+1                                             ODF         99
```

Figure 18-3. 0DF – Example of COMPREL (Cont'd)

```
0DF - DROP FILE.                                                          COMPASS 3.73309.     74/03/07. 16.10.33.      PAGE    14
MAIN ROUTINE.                                                             RTU

   50        3005                              LDD    T5                                                         0DF         100
   51        5415 0110                         STM    DVLA                                                       0DF         101
   53        2000 0113                         LDC    DVL                                                        0DF         102
   55        3515                              RAD    LA                                                         0DF         103
   56        2033 2206                         LDC    3RORF                                                      0DF         104
                             57     RTUA       EQU    *-1                                                        0DF         105
   60                                          EXECUTE  ORF,=      Don't load ORF or assemble any code just set  0DF         106
   60        0200 0533                         RJM.   EXR          up reference for KRONREF                      0DF         107
   62        0355                              UJN    RTUX         RETURN                                        0DF         108


                                    **         PFT - RELEASE PERMANENT FILE                                      0DF         110
                                    *                                                                            0DF         111
                                    *          CALLS  CSF, RTU.                                                  KRA002        9
                                    *                                                                            0DF         114
                                    *          LOADING OF OVERLAYS *0FA*, *ORF* OR *ORP* AND THEIR               KRA002       10
                                    *          PARAMETERS OVERLAYS CODE BEGINNING AT PFTA.                       KRA002       11
                                                                                                                 0DF         117
                                                                                                                 0DF         118
   63        0215 0041              PFT1       RJM    RTU          LOAD AND EXECUTE *0FA* OR *ORP*               KRA002       12
   65        2000 0000                         LDC    0            RESTORE LOAD ADDRESS                          0DF         122
                             66     PFTC       EQU    *-1                                                        0DF         123
   67        3415                              STD    LA                                                         0DF         124
   70        5015 0111                         LDM    PFTA+1       ADJUST CALLING PARAMETERS                     0DF         125
   72        3404                              STD    T4                                                         0DF         126
   73        2000 2000                         LDC    2000         SET OPTION 2 AND RANDOM INDEX                 KRA002       13
                             74     PFTD       EQU    *-1                                                        KRA002       14
   75        3405                              STD    T5                                                         0DF         129
   76        2000 0000                         LDC    0                                                          0DF         130
                             77     PFTE       EQU    *-1          LOWER RANDOM INDEX                            0DF         131
  100        3406                              STD    T6                                                         0DF         132
  101        1512                              LCN    1RP-1RF                                                    0DF         133
  102        5515 0057                         RAM    RTUA                                                       0DF         134
  104        0215 0041              PFT2       RJM    RTU          LOAD AND EXECUTE *0FA*,*ORP*, OR *ORF*        KRA002       15
                                                                                                                 0DF         136
  106        0100 0106              PFT        SUBR                ENTRY/EXIT                                    0DF         137
  110        3013                   PFTA       LDD    CH+3         SET FILE MODE                                 0DF         138
  111        1275                              LPN    75                                                         0DF         139
  112        3405                              STD    T5                                                         0DF         140
  113        3007                              LDD    T3+4         SET FILE WRITTEN STATUS (BIT 1 = 1)           0DF         141
  114        1072                              SHN    -5                                                         0DF         142
  115        1202                              LPN    2                                                          0DF         143
  116        3505                              RAD    T5                                                         0DF         144
  117        3003                              LDD    T3           SET EQUIPMENT                                 0DF         145
  120        1277                              LPN    77                                                         0DF         146
  121        3406                              STD    T6                                                         0DF         147
  122        0215 0315                         RJM    SYS          CLEAR FNT ENTRY                               0DF         148
  124        1412                              LDN    1RP-1RF                                                    0DF         151
  125        5515 0057                         RAM    RTUA                                                       0DF         152
  127        0215 0361                         RJM    CSF          CHECK FOR SPECIAL PERMANENT FILE              KRA002       16
  131        0115 0104                         LJM    PFT2                                                       KRA002       17
                            132     PFTB       EQU    *-1                                                        KRA002       18
                                    *          LJM    PFT1         IF LAST PF ON INTERCHANGABLE DEVICE           KRA002       19
                                                                                                                 0DF         155
                                                                                                                 0DF         156
```

Figure 18-3.   0DF – Example of COMPREL (Cont'd)

```
ODF - DROP FILE.                                          COMPASS 3.73309.     74/03/07. 16.10.33.      PAGE    20
MAIN ROUTINE.                                             CSF

 431          1071                  SHN   -6                                                    KRA002     80
 432          1113                  LMN   FAFT                                                  KRA002     81
 433          0540                  NJN   CSF1      IF NOT FAST ATTACH                          KRA002     82
 434          3011                  LDD   CM+1      CHECK FIRST TRACK                           KRA002     83
 435          3304                  LMD   T4                                                    KRA002     84
 436          0503                  NJN   CSF5      IF NOT SAME TRACK                           KRA002     85
 437          3002                  LDD   T2        SAVE FNT ADDRESS OF FAST ATTACH FILE        KRA002     86
 440          3400                  STD   T0                                                    KRA002     87
 441          0115 0374     CSF5    LJM   CSF1                                                  KRA002     88
 443          3000          CSF6    LDD   T0        CHECK FOR FAST ATTACH FILE                  KRA002     89
 444          0523                  NJN   CSF8      IF FAST ATTACH FILE                         KRA002     90
 445          3007                  LDD   T7                                                    KRA002     91
 446          0417                  ZJN   CSF7      IF NOT LAST FILE ON EQUIPMENT               KRA002     92
 447          3006                  LDD   T6                                                    KRA002     93
 450          5100 0551             ADM.  ESTS      CHECK FOR INTERCHANGABLE DEVICE             KRA002     94
 452          6010                  CRD   CM                                                    KRA002     95
 453          3010                  LDD   CM                                                    KRA002     96
 454          1071                  SHN   -6                                                    KRA002     97
 455          1244                  LPN   44                                                    KRA002     98
 456          1144                  LMN   44                                                    KRA002     99
 457          0506                  NJN   CSF7      IF NOT INTERCHANGABLE                       KRA002    100
 460          1404                  LDN   FNTP      CHECK FOR SPECIAL CALL                      KRA002    101
 461          6010                  CRD   CM                                                    KRA002    102
 462          3010                  LDD   CM                                                    KRA002    103
 463          3257                  SBD   FA                                                    KRA002    104
 464          0711                  MJN   CSF9      IF NOT SPECIAL CALL                         KRA002    105
 465          0115 0360     CSF7    LJM   CSFX      RETURN                                      KRA002    106
 467          3406          CSF8    STD   T6        SET FNT ADDRESS                             KRA002    107
 470          2000 0601             LDC   2RFA      SETUP *OFA* CALL                            KRA002    108
 472          5415 0057             STM   RTUA                                                  KRA002    109
 474          0370                  UJN   CSF7                                                  KRA002    110
 475          2000 0063     CSF9    LDC   PFT1      SET JUMP ADDRESS                            KRA002    111
 477          5415 0132             STM   PFTB                                                  KRA002    112
 501          3015                  LDD   LA                                                    KRA002    113
 502          5415 0066             STM   PFTC                                                  KRA002    114
 504          3074                  LDD   CP        SET OPTION AND RANDOM INDEX                 KRA002    115
 505          1671                  ADN   RFCW                                                  KRA002    116
 506          6010                  CRD   CM                                                    KRA002    117
 507          3014                  LDD   CM+4                                                  KRA002    118
 510          5415 0077             STM   PFTE                                                  KRA002    119
 512          3013                  LDD   CM+3                                                  KRA002    120
 513          1277                  LPN   77                                                    KRA002    121
 514          5515 0074             RAM   PFTD                                                  KRA002    122
 516          0346                  UJN   CSF7                                                  KRA002    123


 517                                RSTR            RESTORE INSTRUCTIONS                        ODF       422
```

All instructions following RSTR will not be relocated

```
 517                                END                                                         ODF       424
```

Figure 18-3. 0DF – Example of COMPREL (Cont'd)

```
                    IDENT  OAV,/REL/VUNX    *                      OAV
                    PERIPM J                                       OAV
   D_M              BASE   MIXEO                                   OAV
                    SST                                            OAV
    0     IRA$      SET    0         EXTERNAL PRESET OF RANOOM ADORESSING OECKS OAV
    1     QUAL$     EQU    1         OEFINE UNQUALIFIEO COMMON OECKS   OAV
    1     REL$      SET    1         DEFINE FULL RELOCATION            OAV
           COMMENT 73/05/05. 73/10/12. VERIFY USER ACCOUNT NUMBER.     OAV
                   COMMENT COPYRIGHT CONTROL OATA CORP. 1973.          OAV


          ***      OAV - VERIFY USER NAME.                             OAV
          *        C.J.MATULE  71/01/11.                               OAV
          *        R.P. ROHRBOUGH     72/09/07.                        OAV


          ***             OAV IS A LOCATION FREE ROUTINE WHICH VERIFIES THAT     OAV
          *        THE SPECIFIED USER NUMBER IS A VALIO ONE. THE VALIOATION      OAV
          *        FILE FOR THE CORRECT FAMILY IS SEARCHEO FOR THE GIVEN USER    OAV
          *        NUMBER AND THE VALIO USER INOEX IS RETURNEO IF FOUND.         OAV
          *        THE ACCOUNT RECORO BLOCK IS ALSO SET UP FOR THE CALLER,S USE. OAV


          **       ENTRY CONOITIONS.                                   OAV
          *                                                            OAV
          *                                                            OAV
          *        (UN - UN+4) = USER NAME.                            OAV
          *                                                            OAV
          *        (CN - CN+4) = FAMILY NAME.                          OAV
          *               = 0 IF NOT AVAILABLE.                        OAV


          **       EXIT CONOITIONS.                                    OAV
```

Figure 18-4. 0AV – Example of COMPRLI

```
0AV - VERIFY USER NAME.                                  COMPASS 3.73309.     74/03/07. 16.09.44.     PAGE     3

*                                                                              0AV
*                                                                              0AV
*              (T1 - T2) = 0 IF THE USER NAME WAS NOT FOUND.                   0AV
*              (T1 - T2) = USER INDEX IF FOUND.                                0AV
*              (T3) = FWA OF ACCOUNT RECORD BLOCK.                             0AV
*              (T4) = 0    UI DOES NOT EXCEED  AUIMX.                          0AV
*              (T4) = 1    UI EXCEEDS  AUIMX.                                  0AV
*              (T5) = FAMILY EQUIPMENT.                                        0AV
```

```
               CTEXT  COMPMAC - PP SYSTEM MACROS.
               CTEXT  COMPRLI - RELOCATABLE OVERLAY MACROS.

               MAXIMUM OVERLAY LENGTH FOR 0AV - INCLUDING BUFFERS.

1673    OAVM   EQU    100*5*2+473  TWO SECTORS OF PROGRAM AND DATA
```

Figure 18-4. 0AV – Example of COMPRLI (Cont'd)

```
OAV - VFRIFY USER NAME.                                   COMPASS 3.73309.    74/03/07. 16.09.44.     PAGE    15
MAIN ROUTINE.                                             VUN                 REL

                                    **      VUN - MAIN PROGRAM.                                    OAV      80
                                                                 MEANS all M-type instruc-         OAV      81
                                                                 tions will be relocated by the    OAV      82
  5       0100 0005          VUN      SUBR             ENTRY/EXIT   remote table.                  OAV      83
                         1   REL$     EQU     1        SET FULL RELOCATION                         OAV1      1
  7       0215 0722                   RJM.    REL,LA   RELOCATE ADDRESSES                          OAV      84
 11       0200 0442                   RJM     IVF      INITIALIZE VALIOATION FILE                  OAV      85
 13       0471                        ZJN     VUNX     EXIT IF FILE NOT AVAILABLE                  OAV      86
 14       0200 0020                   RJM     SUN      SEARCH FOR USER NUMBER                      OAV      87
 16       0366                        UJN     VUNX     RETURN                                      OAV      88
             -M- type will be relocated by remote table


                                    **      SUN - SEARCH FOR USER NUMBER.                          OAV      90
                                    *                                                              OAV      91
                                    *       ENTRY  VALIOATION FILE ATTACHED.                       OAV      92
                                    *              (UN - UN+3) = USER NUMBER.                      OAV      93
                                                                                                   OAV      94
                                                                                                   OAV      95
                                    *       LOCATE PRIMARY LEVEL BLOCK FOR ACCOUNT NUMBER.         OAV      96
                                                                                                   OAV      97
 17       0100 0017          SUN      SUBR             ENTRY/EXIT                                  OAV      98
 21       0200 0120                   RJM     SIB      SEARCH INOEX BLOCK (LEVEL 0)                OAV      99
 23       0716                        MJN     SUN1     IF BAO AOORESS                              OAV     100
 24       0200 0606 will not be       RJM     POS relocated POSITION OISK                          OAV     101
 26       0200 0120                   RJM     SIB      SEARCH INOEX BLOCK (LEVEL 1)                OAV     102
 30       0711                        MJN     SUN1     IF BAO INOEX                                OAV     103
 31       0200 0606                   RJM     POS      POSITION DISK                               OAV     104
 33       2000 0441                   LOC.    BUF-2    REAO OATA BLOCK                             OAV     105
 35       0200 0327                   RJM     RNS      REAO NEXT SECTOR                            OAV     106
 37       5600 0053                   AOM     SUNB     SET FLAG FOR HIT                            OAV     107
                                                                                                   OAV     108
                                    *       RELEASE VALIOATION FILE.                               OAV     109
                                                                                                   OAV     110
 41       3004               SUN1     LOO     T4       OROP CHANNEL                                OAV     111
 42       0200 0446                   RJM     DCH                                                  OAV     112
 44       2000 0044                   LOC     *        SET COMPLETE BIT                            OAV     113
                        45   SUNA     EQU     *-1                                                  OAV     114
 46       6010                        CRO     CM       REAO UP FST ENTRY                           OAV     115
 47       3402                        STD     T2                                                   OAV     116
 50       3614                        AOO     CM+4     SET FILE COMPLETE                           OAV     117
 51       3002                        LOD     T2                                                   OAV     118
 52       6210                        CHO     CM                                                   OAV     119
 53       1400               SUNB     LON     0        FLAG                                        OAV     120
 54       0522                        NJN     SUN6     IF ACCOUNT RECORO TO SEARCH                 OAV     121
                                                                                                   OAV     122
                                    *       SET USER INOEX ANO EXIT.                               OAV     123
                                                                                                   OAV     124
 55       3402               SUN2     STO     T2       SET USER INOEX                              OAV     125
 56       1063                        SHN     -14                                                  OAV     126
 57       3401                        STD     T1                                                   OAV     127
 60       1400                        LON     0        CLEAR UI EXCEEO FLAG                        OAV     128
 61       3404                        STO     T4                                                   OAV     129
 62       3001                        LOO     T1       CHECK UI FOR >  AUIMX                       OAV     130
 63       1237                        LPN     37                                                   OAV     131
 64       1014                        SHN     14                                                   OAV     132
```

Figure 18-4. 0AV – Example of COMPRLI with REL$ Defined

```
                            negative bit for Constant.      This instruction will be relocated via
  65       3302                              LMO   T2       remote table                             0AV
  66       2140 0077                         AOC   -AUIMX                                            0AV
  70       0702                              MJN   SUN3     IF UI NOT >  AUIMX                       0AV
  71       3604                              AOO   T4       SET UI EXCEED FLAG                       0AV
  72       0100 0017             SUN3        LJM   SUNX     EXIT                                     0AV
                                                                                                     0AV
  74       1400                  SUN4        LDN   0                                                 0AV
  75       0357                  SUN5        UJN   SUN2     EXIT WITH NO USER INDEX                  0AV
                                                                                                     0AV
                            *                SEARCH BLOCK FOR ACCOUNT NUMBER.                        0AV
                                                                                                     0AV
  76       0200 0271             SUN6        RJM   SBL      SET LIMIT OF DATA IN BUFFER              0AV
 100       2077 7664             SUN7        LDC   -ARBS*5                                           0AV
 102       3503                              RAD   T3                                                0AV
 103       3201                              SBD   T1       CHECK FOR LIMIT                          0AV
 104       0767                              MJN   SUN4     IF NO VALUE HIT                          0AV
 105       0200 0247                         RJM   CAN      COMPARE ACCOUNT NUMBER                   0AV
 107       0570                              NJN   SUN7     IF NOT EQUAL                             0AV
 110       5003 0003                         LDM   3,T3     will be relocated via remote table       0AV
 112       1277                              LPN   77                                                0AV
 113       1014                              SHN   14                                                0AV
 114       5303 0004                         LMM   4,T3                                              0AV
 116       0356                              UJN   SUN5     EXIT                                     0AV


                            **               SIB -  SEARCH INDEX BLOCK.                              0AV
                            *                                                                        0AV
                            *                ENTRY                                                   0AV
                            *                      (FTDV) = FIRST TRACK.                             0AV
                            *                      DISK POSITIONED.                                  0AV
                            *                      (T6) = TRACK.                                     0AV
                            *                      (T7) = SECTOR.                                    0AV
                            *                      CHANNEL ATTACHED.                                 0AV
                            *                                                                        0AV
                            *                EXIT                                                    0AV
                            *                      (A) < 0 IF ERROR.                                 0AV
                            *                                                                        0AV
                            *                CALLS RNS, CRA, CAN, SBL, SRI.                          0AV
                            *                                                                        0AV
                            *                USES  T1, T3, RI - RI+1.                                0AV
                                                                                                     0AV
                                                                                                     0AV
 117       0100 0117             SIB         SUBR           ENTRY/EXIT                               0AV
 121       2000 0441             SIB1        LDC.  BUF-2    READ NEXT SECTOR                         0AV
 123       0200 0327                         RJM   RNS      READ NEXT SECTOR                         0AV
 125       0200 0271                         RJM   SBL      SET LIMIT OF INDEX ENTRIES IN BUFFER     0AV
 127       1512                  SIB2        LCN   ANWE*5   DECREMENT ENTRY                          0AV
 130       3503                              RAD   T3                                                0AV
 131       3201                              SBD   T1                                                0AV
 132       0620                              PJN   SIB3     IF NOT BEFORE FIRST ENTRY IN BLOCK       0AV
 133       5000 0443                         LDM   BUF                                               0AV
 135       0403                              ZJN   SIB2.2   IF LEVEL - 0                             0AV
 136       1500                  SIB2.1      LCN   0                                                 0AV
 137       0357                              UJN   SIBX     EXIT ERROR FLAGED                        0AV
 140       2000 0000             SIB2.2      LDC.  0                                                 0AV
```

Figure 18-4. 0AV – Example of COMPRLI with REL$ Defined

```
                              **      VUN - MAIN PROGRAM.                                   0AV
                                                                                            0AV
                                                                                            0AV
   5       0100 0005          VUN     SUBR              ENTRY/EXIT                          0AV
   7       0215 0722                  RJM.    REL,LA    RELOCATE ADDRESSES                  0AV
  11       0215 0442                  RJM     IVF       INITIALIZE VALIDATION FILE          0AV
  13       0471                       ZJN     VUNX      EXIT IF FILE NOT AVAILABLE          0AV
  14       0215 0020                  RJM     SUN       SEARCH FOR USER NUMBER              0AV
  16       0366                       UJN     VUNX      RETURN                              0AV


                              **      SUN - SEARCH FOR USER NUMBER.                         0AV
                              *                                                             0AV
                              *       ENTRY   VALIDATION FILE ATTACHED.                     0AV
                              *               (UN - UN+3) = USER NUMBER.                    0AV
                                                                                            0AV
                                                                                            0AV
                              *       LOCATE PRIMARY LEVEL BLOCK FOR ACCOUNT NUMBER.        0AV
                                                        SYSTEXT defined symbol              0AV
  17       0100 0017          SUN     SUBR              ENTRY/EXIT                          0AV
  21       0215 0120                  RJM     SIB       SEARCH INDEX BLOCK (LEVEL 0)        0AV
  23       0716                       MJN     SUN1      IF BAD ADDRESS                      0AV
  24       0200 0606                  RJM     POS       POSITION DISK                       0AV
  26       0215 0120                  RJM     SIB       SEARCH INDEX BLOCK (LEVEL 1)        0AV
  30       0711                       MJN     SUN1      IF BAD INDEX                        0AV
  31       0200 0606                  RJM     POS       POSITION DISK                       0AV
  33       2000 0441                  LDC.    BUF-2     READ DATA BLOCK                     0AV
  35       0215 0327                  RJM     RNS       READ NEXT SECTOR                    0AV
  37       5615 0053                  ADM     SUNB      SET FLAG FOR HIT                    0AV
                                                                                            0AV
                              *       RELEASE VALIDATION FILE.                              0AV
                                                        SYSTEXT defined symbol              0AV
  41       3004               SUN1    LDD     T4        DROP CHANNEL                        0AV
  42       0200 0446                  RJM     DCH                                           0AV
  44       2000 0044                  LDC     *         SET COMPLETE BIT                    0AV
                       45     SUNA    EQU     *-1                                           0AV
  46       6010                       CRD     CM        READ UP FST ENTRY                   0AV
  47       3402                       STD     T2                                            0AV
  50       3614                       ADD     CM+4      SET FILE COMPLETE                   0AV
  51       3002                       LDD     T2                                            0AV
  52       6210                       CWD     CM                                            0AV
  53       1400               SUNB    LDN     0         FLAG                                0AV
  54       0522                       NJN     SUN6      IF ACCOUNT RECORD TO SEARCH         0AV
                                                                                            0AV
                              *       SET USER INDEX AND EXIT.                              0AV
                                                                                            0AV
  55       3402               SUN2    STD     T2        SET USER INDEX                      0AV
  56       1063                       SHN     -14                                           0AV
  57       3401                       STD     T1                                            0AV
  60       1400                       LDN     0         CLEAR UI EXCEED FLAG                0AV
  61       3404                       STD     T4                                            0AV
  62       3001                       LDD     T1        CHECK UI FOR >  AUTMX               0AV
  63       1237                       LPN     37                                            0AV
  64       1014                       SHN     14                                            0AV
  65       3302                       LMD     T2                                            0AV
```

Figure 18-4.   0AV – Example of COMPRLI with REL$ Not Defined

```
 66     2140 0077 negative bit for  ADC   -AUIMX                                    0AV
 70     0702      constant          MJN   SUN3          IF UI NOT > AUIMX           0AV
 71     3604                        AOD   T4            SET UI EXCEED FLAG          0AV
 72     0115 0017         SUN3      LJM   SUNX          EXIT                        0AV
                                                                                    0AV
 74     1400              SUN4      LDN   0                                         0AV
 75     0357              SUN5      UJN   SUN2          EXIT WITH NO USER INDEX     0AV
                                                                                    0AV
                       *            SEARCH BLOCK FOR ACCOUNT NUMBER.                0AV
                                                                                    0AV
 76     0215 0271         SUN6      RJM   SBL           SET LIMIT OF DATA IN BUFFER 0AV
100     2077 7664         SUN7      LDC   -ARBS*5                                   0AV
102     3503                        RAD   T3                                        0AV
103     3201                        SBD   T1            CHECK FOR LIMIT             0AV
104     0767                        MJN   SUN4          IF NO VALUE HIT             0AV
105     0215 0247                   RJM   CAN           COMPARE ACCOUNT NUMBER      0AV
107     0570                        NJN   SUN7          IF NOT EQUAL                0AV
110     5003 0003                   LDM   3,T3          d field defined, the -m- part will be   0AV
112     1277                        LPN   77            relocated by REL via remote table.      0AV
113     1014                        SHN   14                                        0AV
114     5303 0004                   LMM   4,T3                                      0AV
116     0356                        UJN   SUN5          EXIT                        0AV


                       **           SIB -  SEARCH INDEX BLOCK.                      0AV
                       *                                                            0AV
                       *            ENTRY                                           0AV
                       *                  (FTOV) = FIRST TRACK.                     0AV
                       *                  DISK POSITIONED.                          0AV
                       *                  (T6) = TRACK.                             0AV
                       *                  (T7) = SECTOR.                            0AV
                       *                  CHANNEL ATTACHED.                         0AV
                       *                                                            0AV
                       *            EXIT                                            0AV
                       *                  (A) < 0 IF ERROR.                         0AV
                       *                                                            0AV
                       *            CALLS RNS, CRA, CAN, SBL, SRI.                  0AV
                       *                                                            0AV
                       *            USES  T1, T3, RI - RI+1.                        0AV
                                                                                    0AV
                                                                                    0AV
117     0100 0117         SIB       SUBR                ENTRY/ EXIT                 0AV
121     2000 0441         SIB1      LDC.  BUF-2         READ NEXT SECTOR            0AV
123     0215 0327                   RJM   RNS           READ NEXT SECTOR            0AV
125     0215 0271                   RJM   SBL           SET LIMIT OF INDEX ENTRIES IN BUFFER  0AV
127     1512              SIB2      LCN   ANWE*5        DECREMENT ENTRY             0AV
130     3503                        RAD   T3                                        0AV
131     3201                        SBD   T1                                        0AV
132     0620                        PJN   SIB3          IF NOT BEFORE FIRST ENTRY IN BLOCK    0AV
133     5015 0443                   LDM   BUF                                       0AV
135     0403                        ZJN   SIB2.2        IF LEVEL - 0                0AV
136     1500              SIB2.1    LCN   0                                         0AV
137     0357                        UJN   SIBX          EXIT ERROR FLAGED           0AV
140     2000 0000         SIB2.2    LDC.  0             no relocation"."            0AV
               141        SIBA      EQU   *-1                                       0AV
```

Figure 18-4. 0AV – Example of COMPRLI with REL$ Not Defined

```
142        3416                        STD   RI                                                  0AV
143        5115 0147                   ADM   SIBB                                                0AV
145        0470                        ZJN   SIB2.1      IF NOT SET                              0AV
146        2000 0000                   LDC.  0           no relocation "."                       0AV
                     147   SIBB        EQU   *-1                                                 0AV
150        3417                        STD   RI+1                                                0AV
151        0307                        UJN   SIB4                                                0AV
152        0215 0315         SIB3      RJM   SRI         SET RANDOM INDEX                        0AV
154        0215 0247                   RJM   CAN         COMPARE ACCOUNT NUMBERS                 0AV
156        0711                        MJN   SIB6        IF ACCOUNT PAST ENTRY                   0AV
157        0547                        NJN   SIB2        IF NO HIT                               0AV
160        2000 0000         SIB4      LDC   **          d field defined so remote table         0AV
                     161   FTOV        EQU   *-1         FIRST TRACK OF VALIDATION FILE          0AV
162        3406                        STD   T6                                                  0AV
163        0215 0366                   RJM   CRA         CONVERT RANDOM INDEX                    0AV
165        0115 0117         SIB5      LJM   SIBX        RETURN                                  0AV
                                                                                                 0AV
167        5015 0444         SIB6      LDM   BUF+1       CHECK ENTRY                             0AV
171        1704                        SBN   2+ANWE                                              0AV
172        1002                        SHN   2                                                   0AV
173        5115 0444                   ADM   BUF+1                                               0AV
175        1704                        SBN   2+ANWE                                              0AV
176        3101                        ADD   T1                                                  0AV
177        3303                        LMD   T3                                                  0AV
200        0506                        NJN   SIB7        IF NOT LAST                             0AV
201        5015 0460                   LDM   BUF+5*2+3   SET LINKED BLOCK                        0AV
203        5115 0461                   ADM   BUF+5*2+4                                           0AV
205        0506                        NJN   SIB8        IF LINK EXISTS                          0AV
206        5015 0443         SIB7      LDM   BUF         CHECK LEVEL                             0AV
210        0447                        ZJN   SIB4        IF LEVEL-0                              0AV
211        1501                        LCN   1                                                   0AV
212        0352              SIB7.1    UJN   SIB5        RETURN                                  0AV
                                                                                                 0AV
213        5015 0443         SIB8      LDM   BUF         CHECK LEVEL                             0AV
215        0511                        NJN   SIB9        IF NOT LEVEL - 0                        0AV
216        5003 0010                   LDM   5+3,T3                                              0AV
220        5415 0141                   STM   SIBA        SAVE RANDOM ADDRESS OF LAST ENTRY       0AV
222        5003 0011                   LDM   5+4,T3                                              0AV
224        5415 0147                   STM   SIBB                                                0AV
226        5015 0460         SIB9      LDM   BUF+5*2+3   SET READ OF LINKED BLOCK                0AV
230        3416                        STD   RI                                                  0AV
231        5015 0461                   LDM   BUF+5*2+4                                           0AV
233        3417                        STD   RI+1                                                0AV
234        2000 0000                   LDC.  0                                                   0AV
                     235   FTOV1       EQU   *-1                                                 0AV
236        3406                        STD   T6          SET FIRST TRACK                         0AV
237        0215 0366                   RJM   CRA         CONVERT RANDOM ADDRESS                  0AV
241        0750                        MJN   SIB7.1      IF BAD ADDRESS                          0AV
242        0200 0606                   RJM   PDS         POSITION DISK                           0AV
244        0115 0121                   LJM   SIB1        READ BLOCK                              0AV


                           **          CAN -  COMPARE ACCOUNT NUMBER.                            0AV
                           *                                                                     0AV
                           *           ENTRY                                                     0AV
```

Figure 18-4.   0AV – Example of COMPRLI with REL$ Not Defined

19.0 INTRODUCTION

By using the COMCMAC common routine, the CPU programmer can display information on the K display, and receive keyboard input. The CONSOLE macro generates the necessary signal informing DSD that the K display facility is desired.

19.1 CONSOLE A COMPASS MACRO

When the CPU programmer wishes to display on the K display, he uses the CONSOLE A COMPASS macro (Figure 19-1). This causes the display of a central memory buffer.

```
                          24          18        18
A       VFD             | .  KBUF  |  RSCR  |  LSCR |

KBUF    BSSZ      8     KEYBOARD BUFFER

                         10   1    1     47       1
LSCR    VFD             | 0 | S  | F  |   0     | I |

                        LEFT-SCREEN CONTROL WORD

                         10   1    1     47       1
RSCR    VFD             | 0 | S  | F  |   0     | I |

                        RIGHT-SCREEN CONTROL WORD
```

Where:

S = Character size

0 = Small (64 characters/line) lines 12B units apart. (Display is 1000B x 1000B units)

1 = Medium (32 characters/line) lines 24B units apart.

F = Format

0 = Program formatted = after the display is selected, data is output until a zero in byte (0) of a word is encountered or until 1000B words have been output. The data must contain all coordinates

1 = Coded format ("C" format) - The buffer is assumed to be in "C" format (line is terminated when byte (0) contains a zero) and is output until a zero is encountered in byte (0) of the first word in a line, or until 1000B words have been displayed.

Figure 19-1. Console A COMPASS Macro Format

I = Display status

If this is preset to zero (0), it may subsequently be checked for non-zero (which indicates data has been displayed at least once).

Figure 19-1. Console A COMPASS Macro Format (continued)

NOTE*

The KRONOS LOADER will only relocate on 30-bit boundaries. Therefore, unless the CPU program is in absolute mode (COMPASS pseudo op ABS), the COMPASS assembler will flag A with an address error.

To use this display from a relocatable format, the following is used:

define A VFD

24	18	18/Data
0	0	0

```
then code:  SX6     LSCR
            SX1     RSCR
            SX2     KBUF
            LX1     18
            LX2     36
            BX6     X6+X1
            BX6     X6+X2
            SA6     A
```

19.2 DISPLAY BUFFER

The first word of the display buffer is the LSCR or RSCR word. For example:

LSCR VFD

10	1	1	47	1
0	1	0	0	0

This translates as VFD

12	36	12
2	0	0

For small characters, it should be coded

VFD

12	36	12
2	0	0

To further illustrate this format, assume that

LSCR VFD

10	1	1	47	1
0	1	0	0	1

is given. This then translates as VFD

12	36	12
2	0	1

However if small size characters are desired, the line should be coded as VFD

12	36	12
0	0	1

*This is no longer necessary. This loader problem was fixed at level 2.

The user can test bit 0 later in the program to determine if this buffer was displayed at least once.

19.2.1 Display Grid Coordinates

The rest of the line is interpreted as coordinates and BCD data, which can appear in any order. Note that the central program is responsible for supplying coordinates. The user can break the display up into a grid that consists of 51 lines and 64 columns. The spacing between columns is 8 coordinate positions and between lines it is 10 coordinate positions. The area that the central program can use are those lines below line 4 and above line 48.

Think of the display grid in terms of an X and Y axis, where (6000, 7000) is the lower left point of reference and the corners are as shown in Figure 19-2.

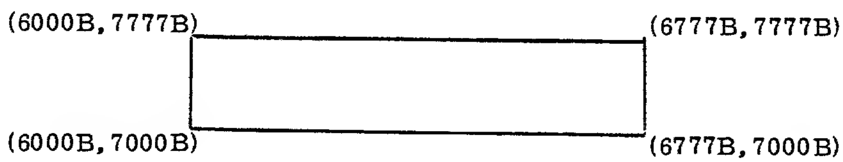


Figure 19-2. Display Grid Corner Reference Points.

However, it is tedious to map characters onto this grid. To simplify this process, the following macro can be used.

```
**      DSL        - DEFINE DISPLAY
*
*       WHERE      X=X COORDINATE
                   Y=Y COORDINATE
                   A=CONSTANT TO BE DISPLAYED

*

 DSL    MACRO      X,Y,A
 B      MICRO      1,6,$A$
 C      MICRO      7,,$A$
        VFD        12/6000B+X*08,12/7756B-Y*10,36/6H"B"
        DATA       H$"C"$
        ENDM
```

According to this macro, the user may invision the grid as shown in Figure 19-3.

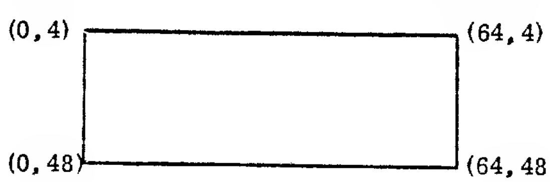


Figure 19-3. Display Grid Displayed

In actuality, the user can use these lines above line 4 and below 48; however, he will overlay the standard K display DSD information which should be avoided.

19.2.2 Display Modifications

If the user wishes to change selected pieces of the display, it is recommended that cells be defined into which BCD information can be stored. Since DSD interprets a zero in byte (0) of any word as an end-of-buffer, these locations must have 5555B in byte (0) in order to display the rest of the buffer.

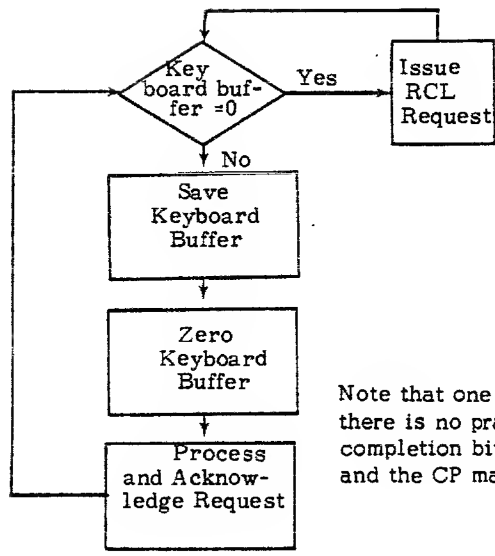
In order to have some parts of the display at a higher intensity, the user merely repaints selected parts of the display. For example, the user can increase the intensity of line 43 by using the DSL macro as shown below:

```
DSL     0,43,data
DSL     0,43,data
```

Flashing of selected parts of the display can be easily coded since any word of zero will act as an end-of-buffer. By placing selected coordinates after a nominal end-of-buffer (word of zeros), the user can set this word to zero, then non-zero according to some counter (see example for a sample of this code).

When receiving information from the keyboard, the buffer (KBUF in this case) is filled with characters when the CR key is pressed. Characters are transmitted to KBUF from the keyboard left justified, 10 per word until exhausted The last word is not filled beyond the final keyboard entry. If one zeroes KBUF prior to receiving entries, the first six bits of zero will signal end-of-information.

A CPU program which communicates with DSD via the keyboard should have a main loop. This could be flow charted as shown in Figure 19-4.



Note that one must go into periodic recall since, there is no practical way to insure that a completion bit will be set with no PP activity and the CP may hang.

Figure 19-4. Sample Main Loop

19.3 DISPLAY PROGRAMMING

Figures 19-5 and 19-6 illustrate an example of a program using the K display. More examples can be found in the routines STAGE, PFS, and MODVAL. Note that these three routines are in (ABS) absolute mode, but the example is in relocatable mode.

```
DISP1     VFD      12/2,36/0,12/2
          DSL      16,5,(TELEX LINE TABLE)
          DSL      2.8,(          CREATED)
DATE1     DATA     2L                                          FILL IN DATE
          DSL      2,11,(         LAST MOD)
DATE2     DATA     2L                                          FILL IN DATE
          DSL      8,14,(OPTIONS AVAILABLE ARE)
          DSL      14,17,(I = INQUIRE)
          DSL      14,19,(C = CREATE)
          DSL      14,21,(D = DELETE)
          DSL      14,23,(M = MODIFY)
          DSL      14,25,(S = SHOW)
          DSL      10,30,(FORMAT IS X, Y)
          DSL      4,33,(WHERE X = OPTION)
          DSL      16,36,(Y = LINE NUMBER - MAX 99)
          DSL      22,38,((EXPANDABLE TO 64000))
          DSL      10,41,(TO STOP RUN TYPE END)
          DSL      0,43,( )       INVALID MESSAGE WILL BE
INV       DATA     2L
INV1      DATA     2L             OVER LAYED HERE
INV1A     DATA     2L             TYPE IN FIELD DISPLAY
FLASH1    DATA     2L             THIS CELL ALTERNATES BETWEEN 0 & 5555B IN
                                  UPPER BYTE IN ORDER TO FLASH INV, INV1
                                  and INV1A
          DSL      0,43,( )
INVB      DATA     2L
INV1B     DATA     2L             DOUBLE INTENSITY
INV1AB    DATA     2L
          DSL      0,43, ( )
INVC      DATA     2L
INV1C     DATA     2L             TRIPLE INTENSITY
INV1AC    DATA     2L
          BSSZ     1                  END OF BUFFER
```

Figure 19-5. Left Screen Display Buffer

If small letters are desired, word DISP1 should be written as:

```
DISP1     VFD      12/0,36/0,12/2
```

Second display buffer

```
DISP1A    VFD      12/2,36/0,12/2
          DSL      20,8, (SST        07)
          DSL      11,16,(TELEX LINE)
          DSL      20,21,(SWITCHING)
          DSL      14,28, (MASTER CONTROL)
          DSL      19,36,(ROUTINE)
          BSSZ     1            END OF BUFFER
```

Figure 19-6. Right Screen Display Buffer

The following routine generates the VFD CMA, CMI, and CMS. It also places the date into the display, displays the buffer, and waits for keyboard input. This routine is only used initially. Thereafter, the next routine, MAJOR, is used.

```
                                COMPASS - VER 2.  73/04/25.  12 53.00.

BDIS        DIS         ,*REQUEST K DISPLAY*
BLANK       DATA        2L
FL4         DATA        200B          FLASHING SPEED BIGGER NO = SLOWER FLASHING
FLASHC      DATA        1
INVALID     DATA        10HINVALID
SPACE       DATA        1H
INBUF       BSSZ        8
DATA        DATA        0
CMA         VFD         24/0,18/0,18/0    24/INBUF,18/DISPLA,18/DIPS1
CMI         VFD         24/0,18/0,18/0    24/INBUF,18/DIPS3,18/DISP2
CMS         VFD         24/0,18/0,18/0    24/INBUF,18/0,18/DISPS
SETK        BSSZ        1
            SX6         DISP1         ADDRESS
            SX1         DISP1A        ADDRESS
            LX1         18
            SX2         INBUF
            LX2         36
            BX6         X6+X1
            BX6         X6+X2         24/INBUF,18/DISPLA,18/DISP1
            SA6         CMA
            SX6         DISP2
            SX1         DISP3
            LX1         18
            BX6         X6+X1
            BX6         X6+X2         24/INBUF,18/DISP3,18/DISP2
            SA6         CMI           These buffers are not shown in this example
            SX6         DISPS
            BX6         X6+X2         24/INBUF/18/0,18/DISPS
            SA6         CMS           This buffer is not shown in this
            DATE        DATE            example get date from system
            SA1         DATE
            BX6         X1
            SA6         DATE1
            SA6         DATE2
            SA6         DATE3         These cells are part of another buffer
            SA6         DATE4            not shown in this example
            MESSAGE     BDIS,1,R      WAIT FOR MESSAGE TO BE DISPLAYED
            CONSOLE     CMA
BEGIN       SA1         INBUF         Ck for input
            NZ          X1,OK
            RECALL
            EQ          BEGIN         LOOP TILL WE GET KEYBOARD ENTRY
OK          BX6         X1
            SA6         IN            argument from calling routine
            EQ          SETK
```

The following routine is called for all subsequent displays of this buffer

```
MAJOR     BSSZ        1
          SA1         INBUF          Last keyboard entry
          BX6         X1
          SA6         INV1A
          SA6         INV1AB         For Flashing
          SA6         INV1AC
*         CLEAR KEYBOARD
          MX6         0
          SA1         CINV
          SA2         CINV1          argument from calling routine
          SA6         INBUF
          SA6         A6+1
          SA6         A6+1
          SA6         A6+1
          SA6         A6+1
          SA6         A6+1
          SA6         A6+1
          BX6         XI
          BX7         X2
          SA6         INV
          SA6         INVB           For flashing
          SA6         INVC
          SA7         INV1
          SA7         INV1B
          SA7         INV1C          For flashing

          CONSOLE     CMA
MA1       BSS         0
          SA1         INBUF
          SA2         A1+1
          NZ          X1,OK11
          RECALL
          SA1         FLASHC
          SA2         BLANK
          SA3         FLASHI
          SA4         FL4
          SX6         XI-1
          SA6         FLASHC
          NZ          X6, MA1
          SX7         X4             Flashing Code
          SA7         FLASHC
          ZR          X3, BLINK1
          MX6         0
          SA6         FLASH1         ZERO OUT AND INDICATE EOB
          EQ          MA1
BLINK1    BX6         X2
          SA6         FLASH1         NON ZERO AND INDICATE NOT EOB
          EQ          MA1            LOOP TILL KEYBOARD RESPONSE
OK11      BSS         0
          BX6         XI
          BX7         X2
          SA6         IN
          SA7         IN+1           ARGUMENTS RETURNED TO CALLING ROUTINE.
          EQ          MAJOR
```

DSD will refresh the last display automatically until a new display is requested The user can modify the display buffer while it is being displayed. This dynamic facility allows the use of flashing code.

The preceding example produces the following left (Figure 19-7) and right (Figure 19-8) screen displays.

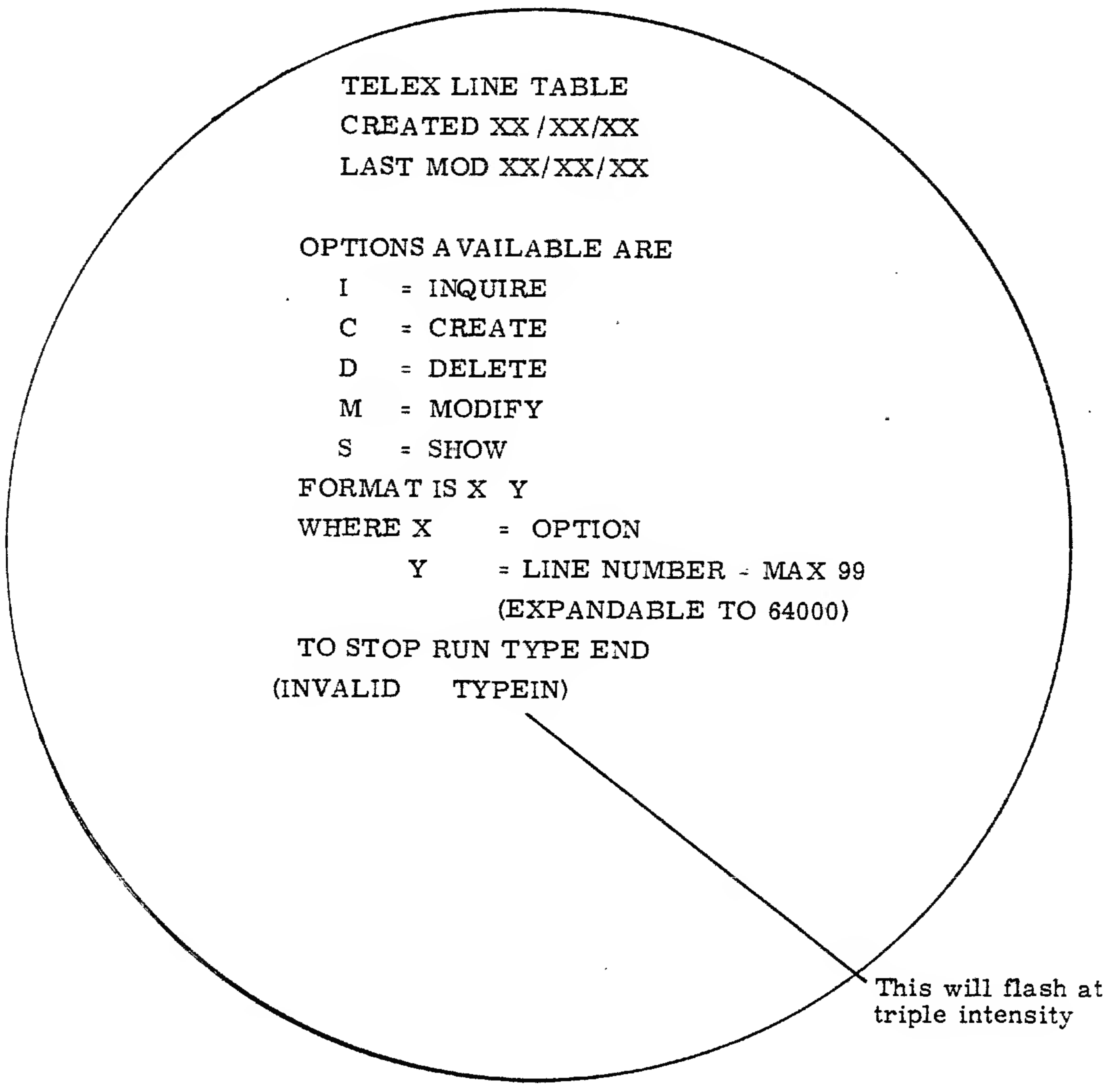


Figure 19-7. Left Screen Display

Figure 19-8. Right Screen Display

The following display on Figure 19-9 is produced by the following program in Figure 19-10.

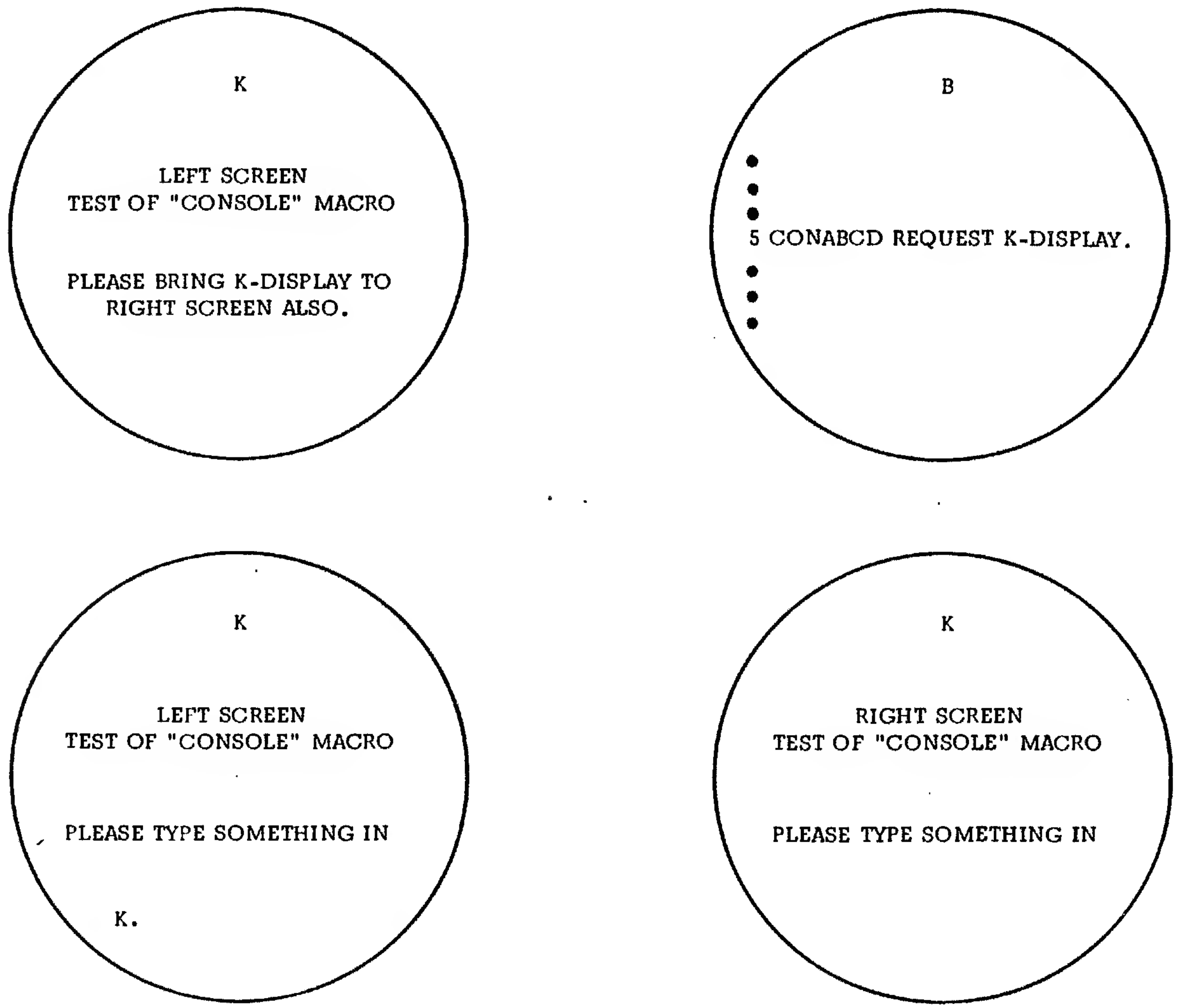


Figure 19-9. Left and Right Screen Display

```
CPM FUNCTION CONSOLE(005).                                      COMPASS 3.74259.    74/12/11. 14.04.01.     PAGE     2

                                         IDENT  CONSOL
                                         ENTRY  CONSOL

    0                                    XTEXT  COMCMAC                                                           L97245

                                 **      DSL - DEFINE DISPLAY LINE
                                 *
                                 *       WHERE  X = X-COORDINATE
                                 *              Y = Y-COORDINATE
                                 *              A = DATA TO BE DISPLAYED

                                  DSL    MACRO  X,Y,A
                                  B      MICRO  1,6,$A$
                                  C      MICRO  7,,$A$
                                         VFD    12/6000B+X*A,12/77569-Y*10,36/6H"B"
                                         DATA   H$"C"$
                                         ENDM

    0                             CONSOL BSS
    0  7110000036 +                      CCNSOLE  DISP       SYSTEM WILL REQUEST *K* DISPLAY

    2                             CPM1   BSS
    2  5110000047 +                      SA1    LS
                 20173                   LX1    59-0
    3  0331000005 +                      NG     X1,CPM2      IF LEFT SCREEN DISPLAYED
                 0100000000 X            RECALL
    4  0400000002 +                      EQ     CPM1         LOOP UNTIL LEFT SCREEN IS DISPLAYED

    5                             CPM2   BSS
    5  5110000101 +                      SA1    RS
                 20173                   LX1    59-0
    6  0331000012 +                      NG     X1,CPM3      IF RIGHT SCREEN DISPLAYED
                 5110000034 +            SA1    BLANKS
    7  10611                             BX6    X1
             5160000057 +                SA6    LS1          DISPLAY MESSAGE TO OPERATOR
   10  0100000000 X                      RECALL
   11  0400000005 +                      EQ     CPM2         LOOP UNTIL RIGHT SCREEN ALSO IS DISPLAYED

                                 *       THE FOLLOWING CODE PUTS OUT A FLASHING MESSAGE ASKING FOR
                                 *       AN OPERATOR TYPE-IN.  THE MESSAGE IS ALTERNATING BETWEEN
                                 *       THE LEFT AND RIGHT SCREENS.

   12                             CPM3   BSS
   12  43600                             MX6    0
             5160000057 +                SA6    LS1
                     66700               SB7    30
   13  6150000012                        SB5    LTYPE

   14                             CPM4   BSS
   14  5117000067 +                      SA1    TYPE+B7      MOVE OPERATOR MESSAGE TO DISPLAY AREA
                 10611                   BX6    X1
   15  5167000060 +                      SA6    LS1+1+B7
                 5167000114 +            SA6    RS1+1+B7
   16  6177000001                        SB7    B7+1
                 0557000014 +            NE     B5,B7,CPM4  IF NOT FINISHED
```

Figure 19-10. Program Display of Figure 19-9. (Sheet 1 of 2)

```
CPH FUNCTION CONSOLE(005).                                         CCNPASS 3.74259.     74/12/11. 14.04.61.      PAGE       3

    17                                CPH5      BSS
    17  5110000037 +                            SA1     KBUF
                  0311000032 +                  NZ      X1,CPH7         IF SONF KFYBOARD INPUT
    20  0100000000 X                            RECALL
    21  5110000034 +                            SA1     BLANKS
                  5120000035 +                  SA2     FLASHC
    22  5130000057 +                            SA3     LS1
                  7262777776                    SX6     X2-1            DECREMENT FLASN COUNTER
    23  5160000035 +                            SA6     FLASHC
                  0316000017 +                  NZ      X6,CPN5         IF NOT TIME TC CHANGE STATUS
    24  7170000310                              SX7     200             RESET FLASH COUNTER
                  5170000035 +                  SA7     FLASNC
    25  0303000030 +                            ZR      X3,CPN6         IF LEFT DISPLAY OFF
                  43600                         NX6     0               LEFT DISPLAY CN....
    26  5160000057 +                            SA6     LS1             SO TURN IT OFF
                  10611                         BX6     X1
    27  5160000113 +                            SA6     RS1             TURN ON RIGHT SCREEN
                  0400000017 +                  EQ      CPH5

    30                                CPH6      BSS
    30  43600                                   MX6     0               TURN OFF PIGNT DISPLAY
              5160000113 +                      SA6     RS1
                         10611                  BX6     X1
    31  5160000057 +                            SA6     LS1             TURN ON LEFT DISPLAY
                  0400000017 +                  EQ      CPN5            LOOP WAITING FOR KEYBOARD INPUT

    32                                CPH7      BSS
    32  7160247021                              ENDRUN

    34  55555555555555555555          BLANKS    DATA    10H
    35  00000000000000000001          FLASNC    DATA    1
    36  00000037 +                    DTSP      VFD     24/KBUF,18/RS,18/LS
                000101 +
                       000047 +
    37                        10      KBUF      BSSZ    8
    47  00020000000000000000          LS        VFD     10/0,1/1,1/0,47/0,1/0
    50  60007612555555551405                    DSL     0,10,(    LEFT SCREEN.)
    52  60007612555555551405                    DSL     0,10,(    LEFT SCREEN.)
    54  60007566240523245517                    DSL     0,12,(TEST OF *CONSOLE* MACRO.)
    57  00000000000000000000          LS1       DATA    0
    60  60007516201405012305                    DSL     0.16,(PLEASE BRING K-DISPLAY TO)
    63  60007472221107102455                    DSL     0,18,(RIGNT SCREEN ALSO.)
    66  00000000000000000000                    DATA    0
    67  60007516201405012305          TYPE      DSL     0,16,(PLEASE TYPE SOMEIHING IN.)
    72  60007516201405012305                    DSL     0,16,(PLEASE TYPE SONFTNING IN.)
    75  60007516201405012305                    DSL     0,16,(PLEASE TYPE SOMFTNING IN.)
   100  00000000000000000000                    DATA    0
                              12      LTYPE     EQU     *-TYPE
   101  00020000000000000000          RS        VFD     10/0,1/1,1/0,47/0,1/0
   102  60007612555555552211                    DSL     0,10,(    RIGNT SCREEN.)
   105  60007612555555552211                    DSL     0,10,(    RIGNT SCREEN.)
   110  60007566240523245517                    DSL     0,12,(TEST OF *CONSOLE* NACRO.)
   113  00000000000000000000          RS1       DATA    0
   114                        12                BSSZ    LTYPE
   126                                          END     CONSOL

                   51000B CM  STORAGE USED                 151 STATEMENTS        21 SYMBOLS
```

Figure 19-10. Program Display of Figure 19-9. (Sheet 2 of 2)

KRONREF AND COMMON DECKS 20

20.0 INTRODUCTION

KRONREF and Common Decks are both pertinent to the system library (OPL). KRONREF is used to locate particular usage of items such as symbol, type, common deck, etc. Common decks are all on the OPL, thus the majority of jobs can be accomplished without special macro definitions.

20.1 KRONREF

KRONREF is useful to the programmer who wishes to locate a particular use of a symbol, type, error flag, common deck, or PP package.

KRONREF generates a cross-reference listing of system symbols used by decks on a MODIFY OPL. A sample of the KRONREF cross-reference listing is given subsequently. The names of programs on the OPL are listed for those decks that reference the following:

- PP direct cell locations defined in lfn_3 or lfn_4
- PP resident entry points defined in lfn_3
- Monitor functions
- Central memory pointers (in low core) defined in lfn_3 or lfn_4
- Central memory locations (in low core) defined in lfn_3 or lfn_4
- Control point area words defined in lfn_3 or lfn_4
- Dayfile message options
- File types and mass storage constants
- Job origin types, queue types, and priorities
- Error flags referenced
- Common deck calls
- PP packages called *1

The KRONREF control card format is:

KRONREF(P=lfn_1, L=lfn_2, S=lfn_3, G=lfn_4)

P=lfn_1	OPL input from file lfn_1. If the P option is omitted or P alone is specified, file OPL is assumed
L=lfn_2	List output on file lfn_2. If the L option is omitted or L alone is specified, file OUTPUT is assumed.
S=lfn_3	System text from overlay lfn_3. If the S option is omitted or S alone is specified, file SYSTEXT is assumed.

*1 Macro EXECUTE nme, = does not generate code to RJM to EXR, but is used exclusively to make a reference for KRONREF to use.

G=lfn_4 System text from local file lfn_4. If G is omitted, system text is acquired as specified or defaulted by the S option. If G alone is specified, local file TEXT is used. Use of the G option overrides any S specification.

As an example, Figure 20-1 gives the references to monitor functions and central memory pointer words.

20.2 COMMON DECKS

The KRONOS common decks are organized in the following classes.

- CP common decks
- PP common decks
- Equivalences
- Table management
- Display routines
- TRANEX common decks

All common decks are on the system library (OPL). Each common deck is identified by the name COMxnnn on the OPL.

where:

x = the letter signifying the type of common deck:

where:

C = CP common deck
P = PP common deck
S = Equivalence type
T = Table management type
D = Display type
B = TRANEX type

nnn = a three-letter designator usually equal to the entry point used in the common deck.

For example, COMCARG is a CP common deck with a subroutine entry point of ARG. This is the argument processing subroutine.

20.2.1 COMCMAC/COMPMAC

Common decks of particular interest are COMCMAC and COMPMAC. These two common decks contain generally used by system-origin jobs. The most frequently used macros are defined in SYSTEXT as CPCOM and PPCOM. Thus, non system-origin jobs (the majority of jobs) can be written without the need for calling a special common deck of macro definitions. The COMPASS pseudo-op SST causes the assembly of either the CPCOM or PPCOM

```
CROSS REFERENCE OF OPL   OPL FILE=OPL   SYS.TEXT=SYSTEXT (KRONOS 2.1-01/AA) 73/08/29.11.41.29 PAGE 6
MONITOR FUNCTIONS
SYMBOL  VALUE  DECK REFERENCES.

AEQM     1   COMDTFN  MTR      1DS
AMSM     2   COMDTFN  CIO      LFM  MTR     PFM   PPR   XSP   1CJ   1DS   1RO   1TA
CCHM     3   COMDDSP  COMDTFN  DSD  MTR     1TO
DCHM     4   COMDTFN  DSD      MTR  PPR
DEQM     5   COMDDSP  COMDTFN  BAT  DIS     LFM   MTR   REC   0DF   1AJ   1CD   1CJ
             1ED      1IO      1TD
DFMM     6   COMDTFN  MTR      PPR
OFEM     7   COMDTFN  DSD      MTR  1IO     1MT
ONEM    10   COMDTFN  DSD      MTR  RMS     1DS
PRLM    11   COMDTFN  MTR      PPR  REC     1SD   CPUMTR
RCHM    12   COMDTFN  DSD      MTR  PPR     1ED
REMM    13   COMDTFN  CPM      DIS  MTR
REQM    14   COMDDSP  COMDTFN  DSD  LFM     MTR   1ED   1IO   1TD
ROCM    15   COMDTFN  CIO      CPM  DIS     LFM   MTR   PFM   SFP   1AJ   1CK   1DS
             1MA      1SJ      1SP  1TA
RPRM    16   COMDTFN  CHD      CPM  DIS     MTR   026   SLL   SMP   1AJ   1DS   1MT
             1RI      1SP      1TA  1TD
RJSM    17   COMDTFN  MTR      SFM  1AJ     1DS   1SP   1TA   2TJ
SCHM    20   COMDTFN  MTR      6DD  6DH     6DI   6DP   6MD
RSTM    21   COMPRSI  COMDTFN  ADC  DAT     CMS   DOG   DS1   MTR   REC   SLL   STL
             WRM      1AJ      1CJ  1IO     ILS   1RI   1RO   1SJ
RSYM    22   COMDTFN  CIO      CLL  EXU     MTR   PPR   SFP   1AJ   1DL
SMSM    23   COMDTFN  DSD      MTR
STPM    24   COMDTFN  DSD      MTR
TGPM    25   COMDTFN  MTR      1DS  1TO
TSEM    26   COMDTFN  CPM      MTR  TLX     1AJ   1DS   1RI   1TA   1TD   1TO
DEPM    27   COMDTFN  MTR
DRCM    30   COMDTFN  CIO      MTR  STL     1MT
SCPM    31   COMDTFN  CPM      MTR
EATM    32   COMDTFN  CPM      MTR  DFA     ORP   1MT   1SP
CPUM    36   COMPMRQ  MTR      PPR  CPUMTR
ABTM    36   COMDTFN  ADC      BAT  CHD     CLL   CMS   CPM   DIS   DOG   DS1   EXU
             LFM      026      PFM  PFU     SFM   SFP   SLL   SMP   TLX   WPM   XSP
             1AJ      1LS      1MA  1MT     CPUMTR
CCAM    37   COMDTFN  REC      RMS  SLL     STL   XSP   1CK   1DS   1MT   1RO   1SJ
             1SP      1TA      CPUMTR
CEFM    40   COMDDSP  COMDTFN  CIO  DIS     DSD   MTR   026   SFP   XSP   1AJ   1DS
             1ED      1MT      1RI  1TD     CPUMTR
OCPM    41   COMDTFN  CHD      DIS  EXU     MTR   026   SMP   1AJ   1LS   CPUMTR
DJSM    42   COMDTFN  CPM      DIS  DSD     026   SFM   SLL   XSP   1CK   1DS   1TA
             CPUMTR
DTKM    43   COMDTFN  CIO      IMS  LFM     PFM   PFU   PPR   REC   RMS   SFM   SFP
             SLL      1SP      0DF  0RP     1CD   1CJ   1CK   1DS   1RI   1RO   1TA
             1TO      CPUMTR
```

Figure 20-1. Cross Reference of OPL.

CROSS REFERENCE OF OPL. OPL FILE-OPL SYS. TEXT=SYSTEXT (KRONOS2.1-01/AA)73/08/29 11.41.29 PAGE 7
MONITOR FUNCTIONS

SYMBOL	VALUE	DECK REFERENCES*										
DPPM	44	COMDTFN	ADC	BAT	CHD	CHK	CIO	CLL	CMS	CPM	DIS	DOG
		DS1	EXU	IMS	LFM	MTR	OUT	026	PFM	PFU	PPR	REC
		SFM	SFP	SLL	SMP	STL	TLX	WRM	XSF	0BF	1AJ	1BA
		1CD	1CK	1DL	1DS	1ED	1IO	1LS	1MA	1MT	1RO	1SJ
		1TA	1TD	1TO	CPUMTR							
ECSM	45	COMDTFN	6DE	CPUMTR								
RCLM	46	COMDTFN	CHD	DIS	DSD	MTR	1TD	CPUMTR				
RCPM	47	COMDTFN	CHD	DIS	EXU	SMP	STL	1AJ	1RI	1RO	CPUMTR	
RDCM	50	COMDTFN	CPM	REC	SFM	1AJ	1BA	1LS	1MT	1RO	1TA	1TD
		CPUMTR										
REWM	51	COMDTFN	CPUMTR									
RJAM	52	COMDTFN	1CJ	CPUMTR								
RPPM	53	COMDTFN	IDS	DOG	DSD	MTR	026	SMP	STL	1CD	1LS	1MT
		1SJ	1TD	CPUMTR								
RSJM	54	COMPRSI	COMDTFN	CPM	DIS	DSD	MTR	026	SFM	SLL	XSP	1AJ
		1CD	1CK	1DS	1LS	1RI	1RO	1SJ	1SP	1TA	CPUMTR	
RTCM	55	COMDTFN	CIO	IMS	MTR	PFM	PFU	PPR	REC	RMS	SLL	XSP
		0BF	1BA	1CJ	1CK	1RO	1TO	CPUMTR				
SFBM	56	COMPFAT	COMPSDI	COMPSFB		CIO	CMS	IMS	PFM	PFU	0FA	0RF
					COMDTFN							
		0RP	1LS	1TA	CPUMTR							
STBM	57	COMPCTI	COMPSII	COMDTFN		PFM	PFU	RMS	ORP	CPUMTR		
					IMS							
UADM	60	COMDTFN	CIO	CPUMTR								
WEHM	61	COMDTFN	CPUMTR									
JACH	62	COMDTFN	MTR	REC	1AJ	1CJ	1RI	1RO	1SJ	CPUMTR		
DLKM	63	COMDTFN	PFM	CPUMTR								
TDAM	64	COMDTFN	CIO	0RF	1DS	1MT	CPUMTR					
TIOM	65	COMDTFN	1MT	CPUMTR								
RTLM	66	COMDTFN	CPM	DIS	1AJ	1DS	1RI	CPUMTR				
LCEM	67	COMDTFN	1AJ	CPUMTR								
CSTM	70	COMDTFN	0DF	CPUMTR								
CKSM	71	COMDTFN	SFP	1AJ	CPUMTR							
MSFM	76	COMDTFN	DIS	0SO	MTR	0DF	1MT	1TA	CPUMTR			
RPLP	1	DIS	DSD	PPR	REC	SLL	STL	1AJ	1CK			
PLDP	2	DIS	MTR	PPR	REC	SLL	1AJ	1DL				
PPCP	2	COMPMRQ	DIS	DSD	MTR	PPR	SET	STL	1TD	CPUMTR		
											DSD1	
DFPP	3	DIS	DSD	MTR	PPR	REC	SET	SFM	1CJ	1CK	1R1	1RO
		CPUMTR										
JBCP	4	COMPRJC	DSD	PFM	SET	1SP	CPUMTR					
FNTP	4	COMPFAT	COMPSAF	CIO	DIS	DSD	LFM	OUT	026	PFM	REC	SET

Figure 20-1. Cross Reference of OPL. (continued)

```
CROSS REFERENCE OF OPL.  OPL FILE OPL  SYS. TEXT=SYSTEXT (KRONOS 2.1-01/AA) 73/08/29 11.41 29 PAGE 9
CENTRAL MEMORY POINTERS
SYMBOL   VALUE   DECK REFERENCES

                 SFM        SLL        XSP  0BF     0DF     0RF     1AJ     1CJ     1CK     1DS     1IO
                 1LS        1RI        1RO  1SJ     1SP     1TA     1TD     CPUMTR
ESTP       5     COMPDTS    COMPFAT    COMPSCA      COMP3XD         BAT     CMS     CPM     CIS     CSO
                                            COMPSDN         COMDDSP
                 EXU        IMS        LFM  MTR     PFM     PFU     PPR     REC     RMS     SET     SFM
                 SFP        SLL        STL  DAV     1CD     1CJ     1CK     1DS     1ED     1IO     1MT
                 1RI        1RO        1SP  1TA     1TD     1TO     6DE     BLANK   CPUMTR          PFCAT
                 PFDUMP     PFLOAD     PFS  RESEX                                           MSI
RCLP       6     REC        SLL
CLDP       7     COMPCLO    CLL        EXU  LFM     REC     SLL     1AJ
SPLP      46     MTR        SLL
PXPP      62     COMPMRQ    PPR        CPUMTR
```

Figure 20-1. Cross Reference of OPL. (continued)

definitions in a routine. In either case, whether the macros used are defined in a common deck or SYSTEXT, the program must also call the appropriate common deck which contains the code to perform the operation requested by the macro. For instance, if the MOVE macro is being used, the program must also call the COMCMVE common deck. This call is generally done by a *CALL card within the program. However, many of the common decks of general (frequent) application are available in relocatable form on the user library SYSLIB. In this case, the call is via an external entry point. For instance, RJ =XCIO= . A list of these relocatable routines is available in Part I Section 3 of the Installation Handbook.

In general, the subroutines available in the common decks have been checked out and optimized and their use is recommended. The S-type common decks contain symbol definitions used in the various subsystems. The display subroutines, D-type, are used by DSD, DIS, and other routines which drive the display console. TRANEX common decks are available on the TRANEX program library file, KTSPL.

Individual copies of the common decks can be assembled with the job.

Example:

```
JOBCARD
ACCOUNT (name, pw)
ATTACH (OPL) Program Library file
MODIFY (Q, CL, Z)/*EDIT, CALLxxx
6/7/8/9
    where,
        xxx = CPU - CP common decks
            = PPU - PP common decks
            = SYS - Equivalence type
            = DIS - Display type
            = TAB - Table type
```

20.2.2 CP Common Deck

The following CP common decks are available on the system OPL.

COMCMAC	-	CPU System Macro Definitions
COMCARG	-	Process Arguments
COMCARM	-	Multiple Word Argument Processor
COMCCDD	-	Constnat to Decimal Display Code Conversion (Up to 10 digits)
COMCCFD	-	Constant to F10.3 Conversion
COMCCIO	-	I/O Function Processor
COMCCOD	-	Constant to Octal Display Code Conversion (Up to 10 digits)
COMCCPM	-	Control Point Manager
COMCCPR	-	Central Processor Abort Recovery Processor (Similar to REPRIEVE in SCOPE 3.4)
COMCDXB	-	Display Code to Binary Conversion

COMCEDT	-	Edit Date or Time from Packed Format
COMCFCE	-	Format Catalog Entry for Output
COMCLFM	-	Local File Manager Processor
COMCMTM	-	Managed Table Macros
COMCMTP	-	Managed Table Processors
COMCMVE	-	Move Block of Data
COMCOVL	-	Load Overlay Processor
COMCPFM	-	Permanent File Processor
COMCPOP	-	Pick Out Parameter
COMCRDC	-	Read Coded Line, -C- Format
COMCRDH	-	Read Coded Line, -H- Format
COMCRDO	-	Read One Word
COMCRDS	-	Read Coded Line to String Buffer
COMCRDW	-	Read Words to Working Buffer
COMCRTN	-	Read Terminal Network Descriptions
COMCSFM	-	System File Manager Processor
COMCSFN	-	Space Fill Name
COMCSRT	-	Set Record Type
COMCSSN	-	Skip Sequence Number
COMCSST	-	Shell Sort Table
COMCSTF	-	Set Terminal Output File
COMCSYS	-	Process System Requests
COMCUPC	-	Unpack Control Card
COMCUSB	-	Unpack Data Block to String Buffer
COMCWOD	-	Convert Word to Octal Display Code
COMCWTC	-	Write Coded Line, -C- Format
COMCWTH	-	Write Coded Line, -H- Format
COMCWTO	-	Write One Word
COMCWTS	-	Write Coded Line from String Buffer
COMCWTW	-	Write Words from Working Buffer

20.2.3 PERIPHERAL PROCESSOR COMMON DECKS

COMPMAC	-	PP System Macros
COMPCHI	-	Redefine I/O Instructions
COMPCDI	-	Clear Permanent File Device Interlock
COMPCIB	-	Check Input Buffer
COMPCLD	-	Search Central Library Directory
COMPCLX	-	Clear Exchange Package
COMPCOB	-	Check Output Buffer
COMPCKP	-	Set Checkpoint Bit In EST Entry
COMPCRA	-	Convert Random Address to Track and Sector
COMPCRS	-	Check Recall Status
COMPCTI	-	Clear Track Interlock
COMPCUA	-	Check User Access
COMPC2D	-	Convert 2 Octal Digits to Display Code
COMPDTS	-	Determine Track Interlock Status
COMPFAT	-	Search for Fast Attach File
COMPGJN	-	Generate Job Name
COMPGTN	-	Generate Terminal Number
COMPIRA	-	Initialize Random Access Processors
COMPMRQ	-	Monitor Request
COMPMSD	-	Mass Storage Processor for 853/854/821/841/814

COMPRCB	-	Read Coded Buffer
COMPRCS	-	Read Control Statement
COMPRJC	-	Read Job Control Word
COMPRNS	-	Read Next Sector
COMPRSI	-	Request Storage Increase
COMPRSS	-	Read System Sector
COMPSAF	-	Search for Assigned File
COMPSCA	-	Set Catalog Address
COMPSDI	-	Set Permanent File Device Interlock
COMPSDN	-	Search for Device Number
COMPSEI	-	Search for End of Information
COMPSFB	-	Set File Busy
COMPSPA	-	Set Pot Address
COMPSNT	-	Set Next Track
COMPSRA	-	Set Random Address
COMPSTA	-	Set Terminal Table Address
COMPSTI	-	Set Track Interlock
COMPUPP	-	Update Pot Pointer
COMPUPS	-	Unpack Statement
COMPVFN	-	Verify File Name
COMPWBB	-	Write Binary Buffer
COMPWCB	-	Write Coded Buffer
COMPWSS	-	Write System Sector
COMP3XD	-	3000 Equipment Driver Subroutines
COMPREL	-	Location Free Overlay Macros
COMPRLI	-	Relocatable Overlay Macros
COMPCHL	-	Redefine I/O Instructions

20.2.4 DISPLAY COMMON DECKS

COMDDIS	-	Display Subroutines
COMDDSP	-	Display Program Routines
COMDSYS	-	Display System Status and Associated Routines
COMDTFN	-	Table of Monitor Functions for Display

20.2.5 SYSTEM COMMON DECKS

COMSACC	-	Account File Equivalences
COMSBIO	-	Batchio Equivalences
COMSCIO	-	CIO/Driver Equivalences
COMSDSL	-	Deadstart Load Parameters
COMSEXP	-	EI200 Tables and Constants
COMSJOT	-	Job Output Equivalences
COMSJRO	-	Job Rollout Equivalences
COMSLDR	-	CPU Program Loading Equivalences
COMSMSP	-	Mass Storage Processing Equivalences
COMSMTR	-	MTR/CPUMTR Equivalences
COMSMTX	-	Magnetic Tape Executive Equivalences
COMSNET	-	Terminal Network Equivalences

COMSPFM	-	Permanent File Equivalences
COMSPFS	-	Permanent File Supervisor Equivalences
COMSPFU	-	Permanent File Utilities Equivalences
COMSPRO	-	Profilo Record Equivalences
COMSREM	-	TELEX System Parameters
COMSRSX	-	Resource Executive Equivalences
COMSSFS	-	Special System File Macros and Equivalences
COMSSSJ	-	Special System Job Parameters
COMSTDR	-	Terminal Driver Equivalences

20.2.6 TABLE COMMON DECKS

COMTBCD	-	Display Code to BCD Table
COMTC29	-	COBOL 029 BCD to Display Code Table
COMTDPC	-	BCD to Display Code Table
COMTF29	-	FORTRAN 029 BCD to Display Code Table
COMTS29	-	SNOBOL 029 BCD to Display Code Table

20.2.7 SPECIAL PURPOSE COMMON DECKS

COMLRUN	-	Run Library Communication Definitions
COMMMSE	-	Mass Storage Error Processor

20.3 COMMON Decks Can Be Called in Three Ways.

1. *CALL.

If the program using a common deck is on a PL and the COMMON decks are on this PL or some other PL available to MODIFY, then the program can use the *CALL directive. See the MODIFY reference manual.

Example. DECK1 is on PL MYPL and the COMMON decks are on OPL.

```
JOB
ACCOUNT
GET,MYPL.
ATTACH,OPL.
MODIFY (Z)/*OPLFILE,MYPL/*EDIT,DECK1
```

where DECK1 contains:

```
*CALL,COMxnnn.
```

2. XTEXT

If the program is not on a PL then the pseudo op XTEXT can be used. The PL containing the COMMON decks must be available.

Example. DECK1 is not on a PL, but the COMMON decks are on the PL OPL.

```
JOB
ACCOUNT
ATTACH,OPL.
COMPASS.
7/8/9
                IDENT           DECK1
                  :               :
OPL             XTEXT           COMxnnn
                  :               :
                END             DECK1
```

3. If the COMMON deck is executable code, and it has been assembled and placed on the system then the user can just external it and the loader will load it in with the user's deck.

Example.

```
IDENT           DECK1
  :               :
RJ            = Xnnn
  :               :
END             DECK1
```

Note that the entry point for all executable COMMON decks is the last three characters of the name.

This method is of course limited to executable decks, which are a part of the system file.

ON—LINE DIAGNOSTIC ROUTINES 21

21.0 INTRODUCTION

The DSD command MAINTENANCE is the same as AUTO, but additionally assigns several maintenance routines at pool processor control points and CPU priorities (refer to the KRONOS 2.1 Operator's Guide for more information on the command..

This section highlights each of these routines. In addition, a routine to test ECS is available from the SCOPE 3.4 Operating System, but is not available with released KRONOS systems. This routine, EC2, compares ECS before and after multiple ECS read and writes. More information on the specific routines can be obtained from the SMM CE operators manual.

21.1 CENTRAL MEMORY TESTER – MY1

MY1 compares memory before and after a write/read. MY1 loads under SMM. The area to be tested is defined by the field length as found on the job card. A0 is set to the field length and defines the upper limits.

This test checks central memory by setting each location from 200 up to the field length to its relative value. It then does five read-backs of each location. The data read back is held in X1 through X5 and is matched against X0, the current test address. It will accumulate and hold all the error bits in X7, store the error accumulations back into memory, read it up and check for zero. It also checks X7 for 0 prior to storing.

At the end of one sweep of memory, the test will then use the complement of the relative address and the check is repeated.

In the event of an error stop, X7 will be holding the accumulation of error bits and X2 through X5 should match X0. Either in the true state or complement form. If no error was indicated in X1 through X5, the error occurred in X1 and was lost when the accumulation check-read was done. If this is the case, the error bits in X7 equal the error bits that occurred in the first X1 read. If the error loop is entered, a DCP entry and recall of the test is necessary to resume operations, or reset P to 3, automatic in KRONOS 2.1.

21.2 RANDOM INSTRUCTION GENERATOR – RAN

RAN randomly creates instructions, sets all memory references within FL, and runs the created job as a subroutine. RAN is a program that generates a set of 10_B random numbers, removes all the jump instructions from this set, and runs it as a subroutine. Passes are inserted in place of 30-bit instructions which occur in the last parcel. All writes and reads are restricted to specific areas.

In order to check the results, a slow loop is also generated. This loop has the same instructions but contains only one instruction for every two words of passes. The B and X registers are loaded with random numbers and the A registers are set to known values before each pass. The slow loop is run first, the results of the registers stored, then the fast loop is run and the results compared. If the results compare, the fast loop is run and compared 14 more times. Providing no error occurs, the fast loop will have been run and compared 15 times for each set of random numbers before a new set is generated.

When an error occurs, the loops will be shortened by one 60-bit word and the test rerun. If it still fails, the loops will be shortened again and the test rerun again and so on until the test doesn't fail. When this happens, the last word removed is replaced and the program halts.

21.3 DIAGNOSTIC MAINTENANCE – ALS

ALS is the same as RAN, except its primary purpose is to test the stack registers and the scoreboard's ability to handle instructions at a faster rate than that possible when not issuing the stack. ALX is a modified ALS which checks a "store after store" operation. ALS/ALX must not use a field length of less than 1500.

Error detection is achieved by executing the same instruction sequence with the same initial register contents twice. The first pass through the instructions is terminated by an 04 jump instruction back to the beginning to achieve execution of all instructions from within the stack. The second pass is terminated by an 02 jump instruction back to the beginning to keep the instructions from being executed from out of the stack registers. Answers are compared and an error stop occurs if they disagree.

Only 03-07 branch instructions are included since 01 and 02 instructions will not branch "in stack." All jumps are to the current address plus one. The branch instruction/instructions will get into the stack only if the branch is not taken on the first pass through the sequence.

All increment reads and writes are to the same address in memory-address 000177. This address is cleared prior to the execution of the instruction sequence.

As part of the initial operands, Register B1 is preset to the loop count of 2 for execution of the instruction sequence. Register B2 is set to 000177 to be used as the increment read and write address. Neither B1 or B2 are used as result registers.

21.4 MODIFIED VERSION OF RAN – FST

FST is a modified version of RAN. The modifications include the optimization of all generating and checking routines and the addition of a new option. Random instructions and operands used are the same as those used in RAN. FST will execute 347362_B passes in 1000_B seconds compared to 230011_B passes for RAN.

The random number generator is at 240-247. The 10_B numbers are saved at 340-347. Passes are inserted for all instructions which occur in the last parcel and for all branch instructions. All writes and reads are restricted to specific areas.

A slow loop is generated from the fast loop instructions with only one instruction for every two words of passes.

The B and X registers are loaded with random numbers and the A registers are set to known values before each pass. The slow loop is run first, the results of the registers stored, then the fast loop is run and the results compared. If the results compare, the fast loop is run and compared 31 more times. Providing no error occurs the fast loop will have been run and compared 32 times for each set of random numbers before a new set is generated.

When an error occurs the loops will be shortened by one 60-bit word and the test rerun, if it still fails the loops will be shortened again and the test rerun again and so on until the test doesn't fail. When this happens, the last word removed is replaced, the pass count entered in the dayfile, and the program halts.

21.5 SIMULATOR – CT3

CT3 simulates a randomly generated set of instructions. The simulator executing differently than the machine loop constitutes an error.

21.6 CENTRAL PROCESSOR TEST 1 – CU1

CU1 tests the central processor control hardware and the central processor functional units, etc. Test of the control hardware checks the real flat settings and the unit reservations. The tests of the functional unit hardware check the arithmetic operations performed in the functional unit for a number of fixed operands.

CU1 needs a basic field length of 10000. However, the last is for the branch unit and utilizes all available field length. If a greater field length is to be used, both A0 and the field length should be set to the new value.

22.0 INTRODUCTION

CHECKPOINT – RESTART is composed of two CP routines, CHKPT and RESTART, which make use of the Special Entry Point (SEP) system described in Section 5. A user must be familiar with this system. The SEP allows these routines to access the privileged file DM*.

By use of a control card call, macro call, or RA+1 request, the user can checkpoint a programs progress for later restart.

By using the RESTART control card, the user can restart a job from any point in the part that he previously checkpointed.

All calls and the use of these routines are described in Section 11 of the KRONOS 2.1 Reference Manual.

22.1 CHECKPOINT FILE

The checkpoint file is one long file, consisting of a series of checkpoint records. Each checkpoint dump is separated by an EOR, a checkpoint control word, and another EOR. An EOI terminates the entire file. A multi-checkpoint file is shown in Figure 22-1.

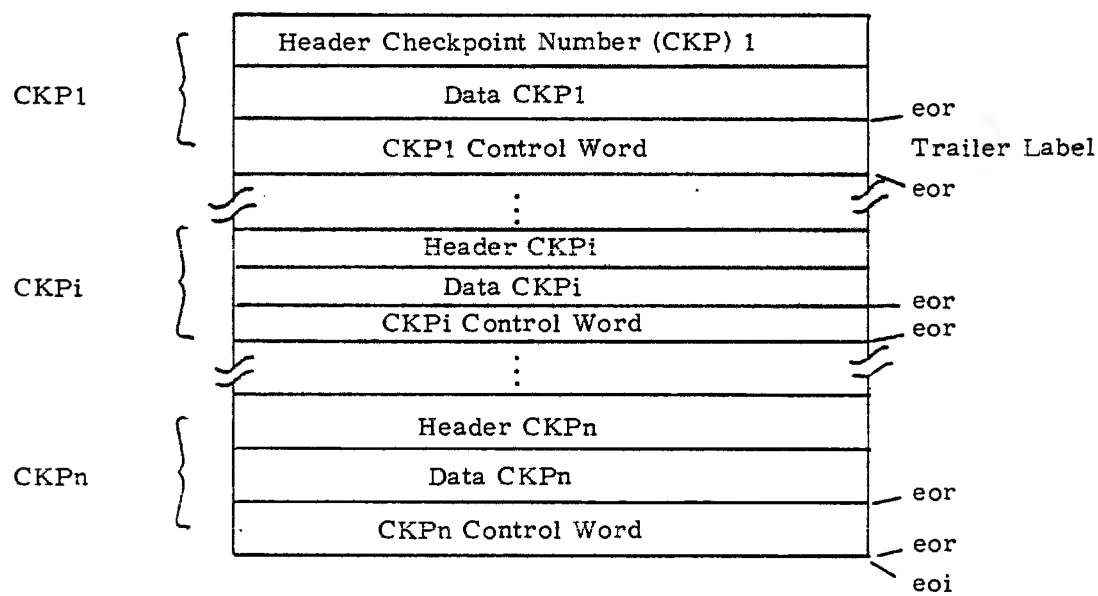


Figure 22-1. A Multi-Checkpoint File

There may be one CKP or many CKPs on the file. If two files are defined, the CKPs will alternate on the files (refer to Section 11 of the KRONOS 2.1 Reference Manual). The files must be requested with the CK or CB option on the REQUEST card, LABEL card, or the ASSIGN card.

There are five parts to each CKP dump (one large record).

1. The header word
2. The file table
3. A copy of each of the files requested
4. A copy of the DM* file of the requesting job
5. A control word (trailer label) embedded between two EORs.

The file is written in control word blocks, using the READW and WRITEW macros. Buffers are always filled before transferring to disk, except for the final control word. Buffers are 1000B words in length which is 10 disk PRUs or 1 tape PRU. Therefore, there are no short PRUs and no EOR, EOF, or EOIs except on the control word block.

In order to indicate the EOR, EOF, and EOIs which occur in the data, a series of control words are used. These control words are:

1. 10002B - header
2. 20NNNB - file table

NOTE

> The following control words indicate that an EOR, EOF, or EOI follows the nnn words of data. The 3xxxx indicates that this is the file copy section.

3. 30nnnB - Start of a block which contains no EOR, EOF, or EOIs.
4. 31nnnB - An EOR occurs at the end of the next nnn words.
5. 32nnnB - An EOF occurs at the end of the next nnn words.
6. 33000B - EOI flag. No data may occur directly before this flag.

NOTE

> The following control words indicate that an EOR, EOF, or EOI follows the nnn words of data in the DM* file.

7. 40nnnB - Start of a block which contains no EOR, EOF, or EOI.

8. 41nnnB - EOR flag

9. 42nnnB - EOF flag

10. 43000B - EOI and end of DM* file

11. 50000B - End of CKP dump

Each CKP dump is one record followed by a control word. Each block on the file is nnn+1 words in length, where nnn is the number of data words preceding this indicator. The maximum physical block size is 1000B words or 777B+1 words. nnn will vary due to EOR, EOF, and EOI occurring in the data. Figure 22-2 shows the format of one CKP file.

The following 15 steps define entries shown in Figure 22-2.

1) date - date CKP file was written

2) time - time CKP file was written

3) CKP mode - is the CKP file a CK (single) or CB (both) id type file

4) CKP number - sequential number of this CKP. I.e., first time CKP called is 1, second time is 2, nth time is n.

5) jobname - job name of job requesting CKP

6) filename - name of a file to be checkpointed

7) WW - job origin or control code from FNT

8) file type - FNT file type, i.e., INPUT, LOCAL, PERMANENT, COMMON, OUTPUT, PUNCH, etc.

9) copy type - portion of file actually copied, as discussed in Section 11 of the KRONOS 2.1 Reference Manual. Unless otherwise specified by the user, files are copied according to their position and type of operation (read or write) prior to the CKP request. The codes are:

   0 - BOI to present position

   1 - present position to EOI

   2 - entire file

   3 - unused

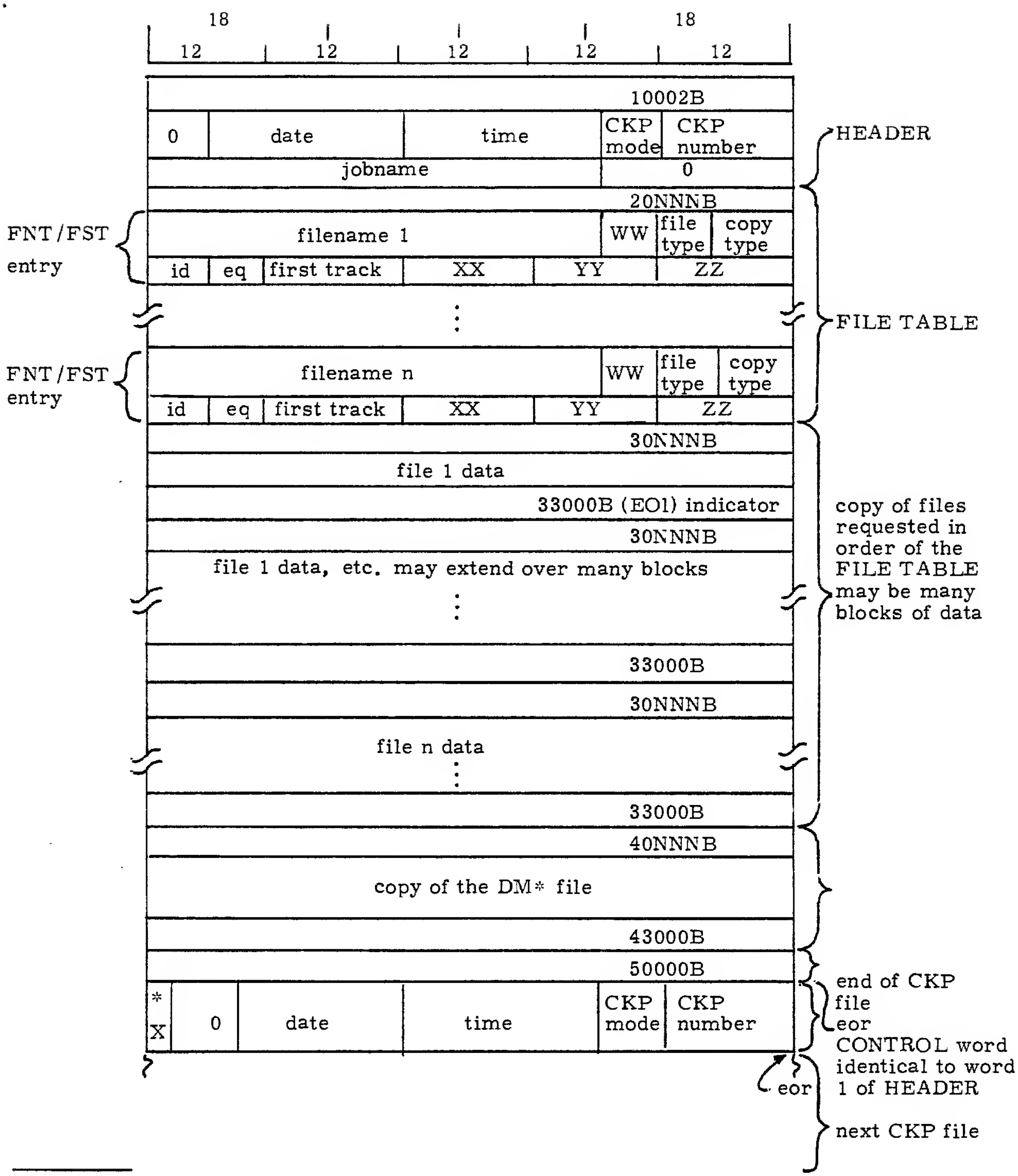
   4 - no copy of file on CKP file.

10) id - from FST

11) eq - from FST

12) first track - if mass storage, then from FST
    if tape, then = "MT"

13) XX - 0 or current track



*Bit X is set if this is the last CKP dump on the file and will be followed by an EOI PRU.
X is in bit 59.

Figure 22-2. CKP Format

14) YY - current sector, field length or format
if tape, then XXYY = block number

15) ZZ - last status from FET

NOTE

10 through 15 are standard FST, except for MT and block number.

Figure 22-3 illustrates how the checkpoint file looks assuming a job has the following characteristics:

1) FL = 2600B, CPA = 200B. So DM* file consists of 200 (CPA) + 4 (FNT/FSTs) + 2600 (FL) = 3004B words.

2) Two files imply 4 words of FNT/FST information.

   File 1 consists of: BOI, 1500B words, EOR, 100B words, EOF, 2001B words, OER, 170B words, EOR, EOF, EOI.

   File 2 consists of: BOI, 100B words, EOR, 1000B words, EOR, EOI.

3) Assume this is a non-terminal job.

22.2 CHECKPOINT – CKP

CHKPT is a CP routine which must reside either in the RCL or be disk resident (CLD - System). CHKPT can be initiated either by an operator command, a control card call, a macro call, or by a SCOPE 3.4 product set call (See Figure 22-4).

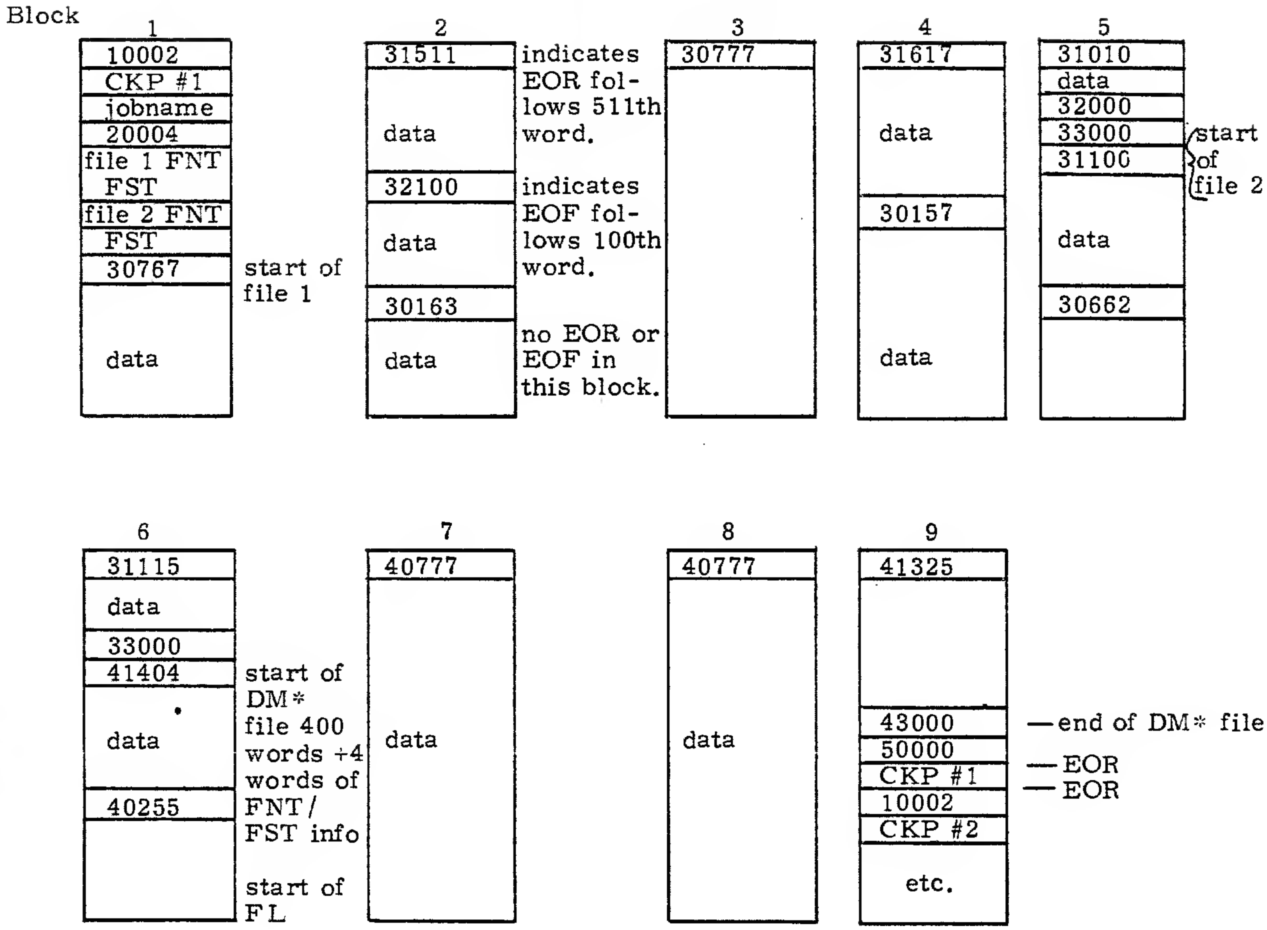
CHKPT has special entry point status. (Refer to Section 5.) CHKPT uses the following SEPs DMP=, SSJ=, and RFL=.

If CHKPT is called by a control card, 1AJ will find that it has an SSJ=, and a DMP= entry point. 1AJ sets up SPCW, SEPW, and the CPA. IRO is called to create the DM* file. Since DMP= is equivalenced to zero in CHKPT, all of central memory is saved on DM*. 1AJ places the arguments from the control card into RA+ARGR and sets RA+PGNR accordingly during the load of CHKPT. Then control is passed to CHKPT.

If CHKPT is called by a macro, an RA+1 request is made to CHKPT. This request is handled by SFP. Therefore, it is necessary for SFP to be an entry point in CHKPT. (See Rule 4 from the flow of an SEP request, Section 5.)

If CKP is called via a SCOPE 3.4 product set, such as FORTRAN or COBOL, an RA+1 request is made and the parameter list, if one is specified, is set up the same as in the macro call.

(All values are in octal)



DM* file is identical to standard rollout file. See Section 5 for DM* file format.

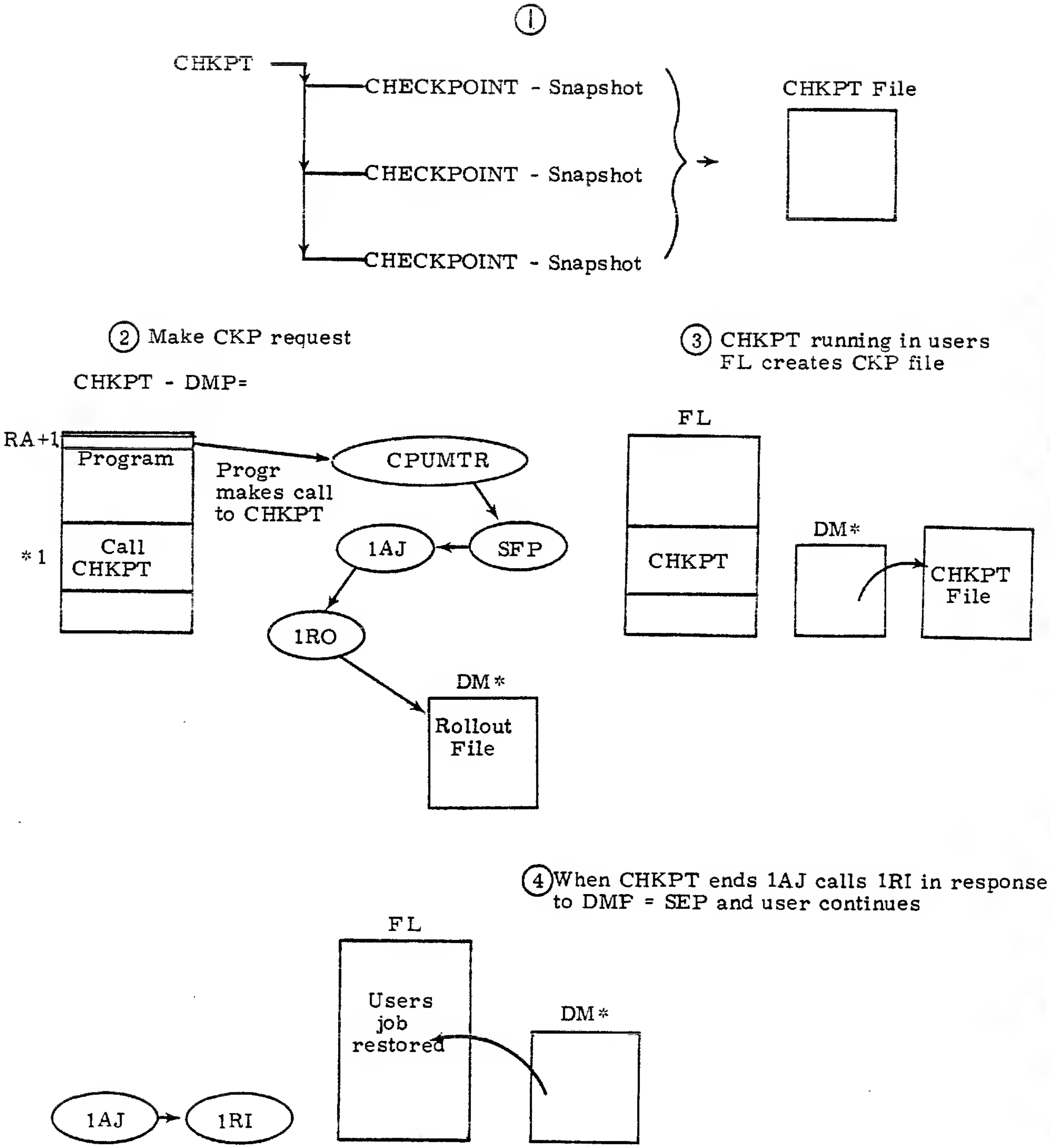
Figure 22-3. Checkpoint File Structure

The RA+1 request is processed by CPUMTR, which places the call into the IR of some available PP.

				12	18
IR =	C K P	CP number	0	n	PBA

where: n = number of parameters

PBA = FWA of parameter list



*1 - creates an RA+1 call.

Note

Refer to Figure 22-7 for RESTART.

Figure 22-4. CHECKPOINT Overview

Since PP resident does not find CKP in either the RPL or in the PLD, it calls SFP.

SFP finds CKP as one of its special processors. The SFP overlay 2SG (SRP - Special Request Processor) sets up SPCW from the IR.

	18	1	1	1	1	2		12		18
SPCW =	CKP	0	0	0	0	0		0	0	PBA

SFP exits normally and 1AJ finds SPCW set. It loads CHKPT, which has the entry point CKP, and sets RA+PGNR=0 to indicate a non-control card call. Since a DMP=SEF has been indicated in the CLD, 1AJ calls 1RO.

1RO finds the PBA not equal to zero and gets a 20B word block from central memory whose FWA is PBA. It creates a full DM* file and then stores the 20B word block in RA+SSPR+1 through RA+SSPR+20. 1AJ sets up SEPW, and any priorities indicated by the SSJ= (in the case of CHKPT, there are no special priorities), stores the IR in RA+SSPR, and passes control to CHKPT at the entry point CKP. If more than a 20B word parameter is passed (CHKPT can be passed up to 200), CHKPT has to read it from the central memory portion of DM*.

CHKPT sets up the CKP file using the DM* file. The SSJ= SEP is superfluous for CHKPT, since SSJ= equals SSJL and SSJL is a 5-word block of zeroes. CHKPT does not require any special priorities, extra FL, or privileges, except access to the DM* file. If the time limit runs out, CKP will be aborted. It is up to the user to ensure that the time limit is adequate for all his needs, including checkpoint time.

Note that the preset routine in CHKPT is overlayed by the buffers, since it ORGs at IBUF. In addition, RFL= is equated to the last word of CHKPT, which is necessary for an SEP using the SSJ= entry point.

Table 22-1 lists some of the common decks used and the buffer assignments.

Figures 22-5 and 22-6 are flowcharts detailing the CHKPT main loop and preset routine.

TABLE 22-1. CHKPT BUFFER ASSIGNMENT/COMMON DECKS

Load Address	Common Decks		
662		CTEXT COMCCDD - Constant to decimal display code conversion	
674		CTEXT COMCCIO - I/O function processor	
706		CTEXT COMCCPM - Control point manager processor	
712		CTEXT COMCDXB - Display code to binary conversion	
727		CTEXT COMCLFM - Local file manager processor	
740		CTEXT COMCMVE - Move block of data	
757		CTEXT COMCRDO - Read one word	
1002		CTEXT COMCRDW - Read words to working buffer	
1116		CTEXT COMCSYS - Process system request	
1150		CTEXT COMCWTO - Write one word	
1166		CTEXT COMCWTW - Write words from working buffer	
		Buffer Assignments	
		USE	BUFFERS
1270	BUF	EQU	*
2270	IBUF	EQU	BUF+BUFL
4271	OBUF	EQU	IBUF+IBUFL
6272	SBUF	EQU	OBUF+OBUFL
6673	TBUF	EQU	SBUF+SBUFL
7674	RFL=	EQU	TBUF+TBUFL

22.3 RESTART

RESTART is a CP routine which must reside either in the RCL or be disk resident. Whereas CHKPT writes DM* onto the CKP file, RESTART restores the contents of the files copied to the CKP file and causes 1RI to restore the CPA and FL from the DM* file. (see Figure 22-7.)

RESTART has the SEP DMP=. When 1AJ loads RESTART, it notes that SEP is active from the CLD or RCL entry point word with the SEP (bit 59) set. It calls 1RO which will create a DM* file. Since DMP= is equated to 45000B in RESTART, it creates an empty DM* file.

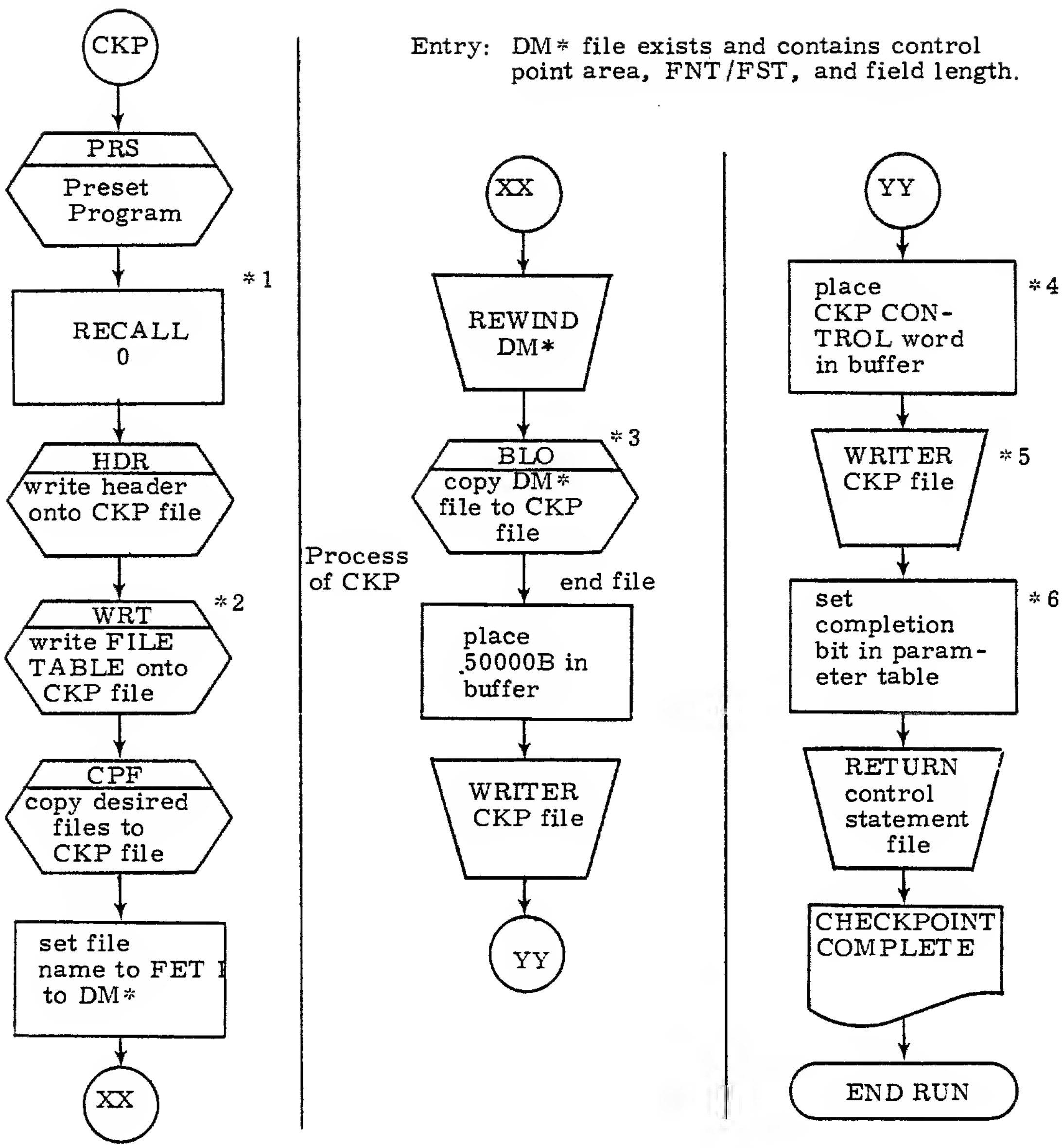
Therefore, DA = | R | O | S | C | F | U | FL (12) | where:

R = 1 RESTART roll-in

C = 1 Create empty DM* file

U = 1 Create DM* as an unlocked file

FL = 0 is ignored since bit C is set.



*1 Wait for any I/O initiated by PRS to complete.

*2 Use GETFNT macro which calls LFM to return a list of all FNT/FSTs assigned to this control point.

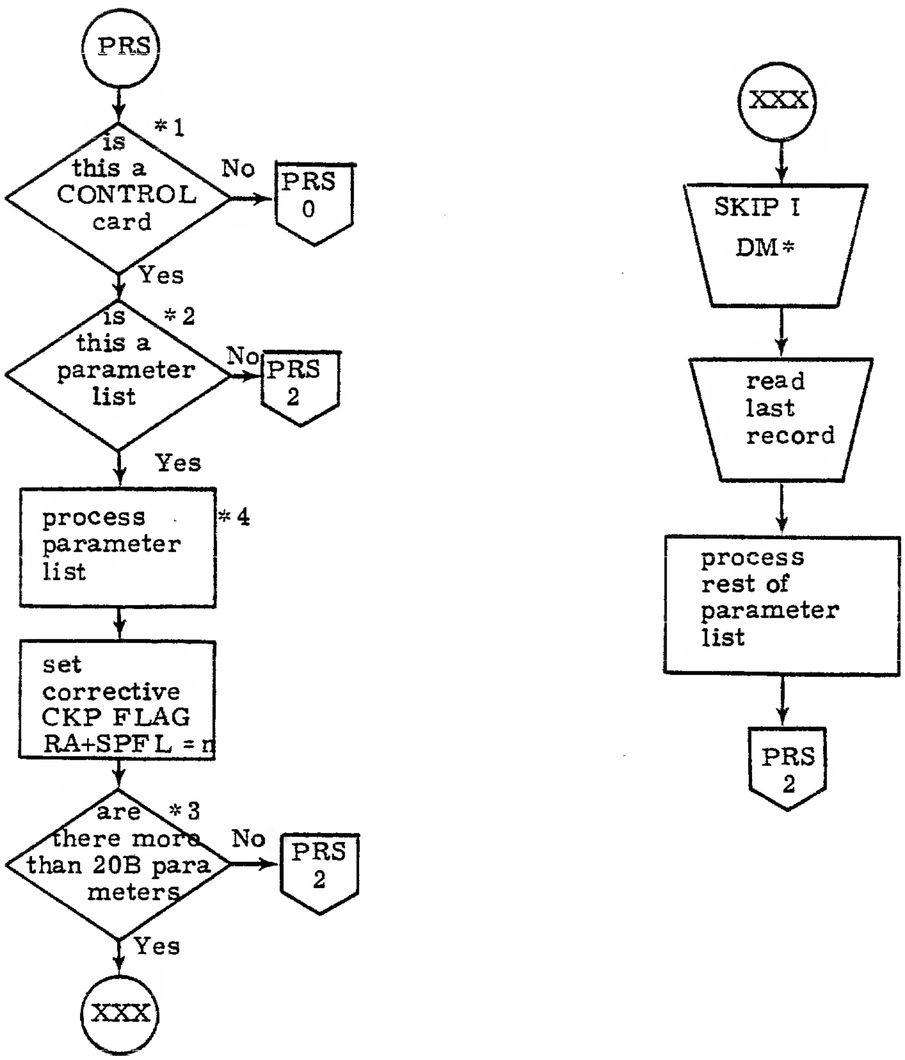
3 See DMP= in Section 5.1.2 SEP. Format of DM file is CPA, FNT/FST, job field length. Copy complete DM* file.

*4 Copy of header word and only word in the buffer.

*5 Now CKP control word is embedded in EORs.

*6 SPRR+1 also backspace file so trailer can be read by next CKP call.

Figure 22-5. CKP - CHECKPOINT (Main Loop)



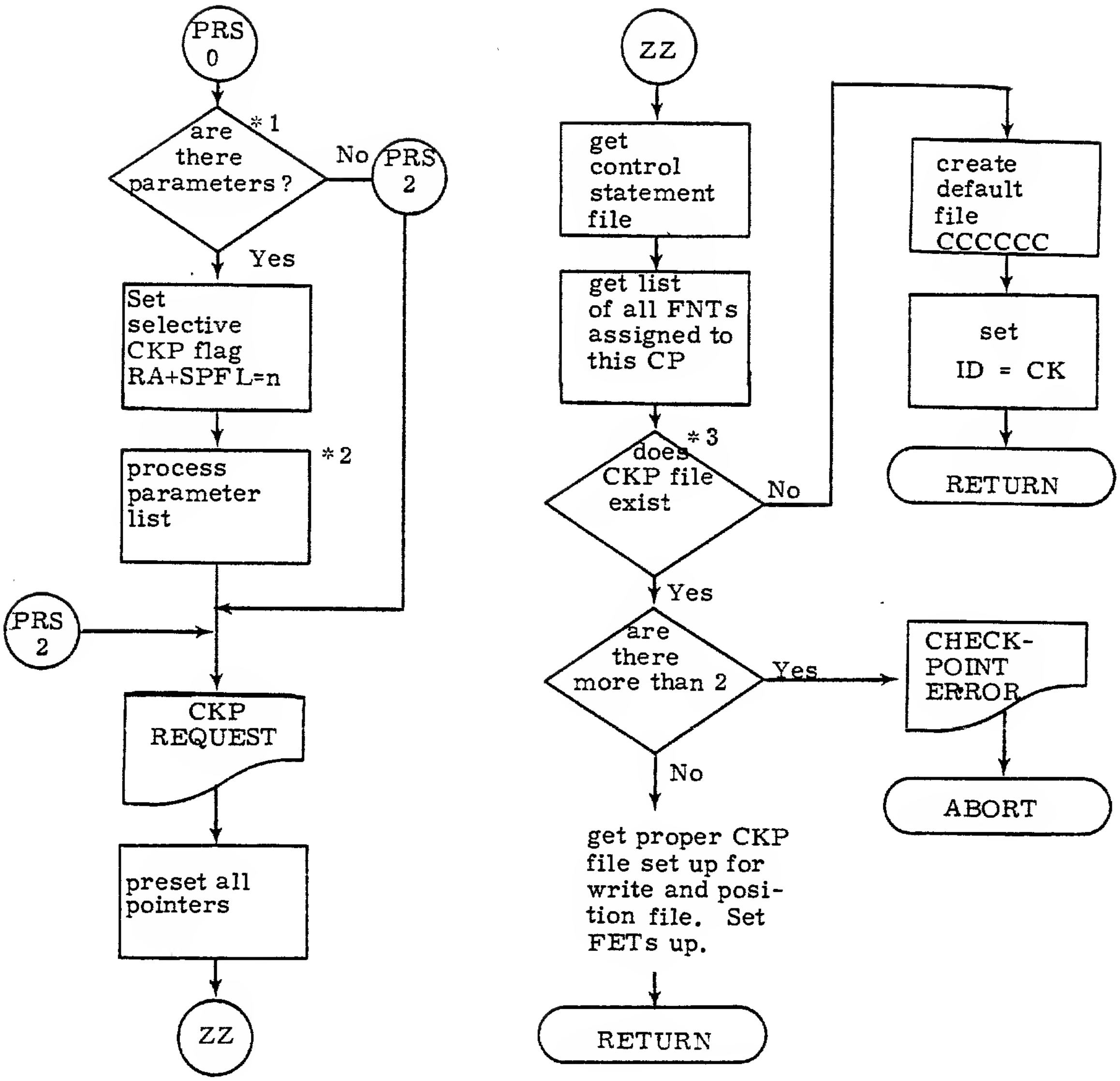
*1 Is (RA+PGNR) not equal zero?

*2 Is (RA+SPPR) lower 18-bit (i.e., n from IR) not equal 0?

*3 Is n > 20.

*4 Parameters = file names are placed in block PAR for use by WRT and CPF to get just selected files onto CKP file; only 77B parameters are allowed.

Figure 22-6. PRS - CHECKPOINT (Preset)



*1 Is RA+ACTR not equal zero.

*2 Maximum 63B parameters on a control card. See footnote 4 on previous page.

*3 See if any files local to job have type CK or CB.

Figure 22-6. PRS - CHECKPOINT (Preset) (Continued)

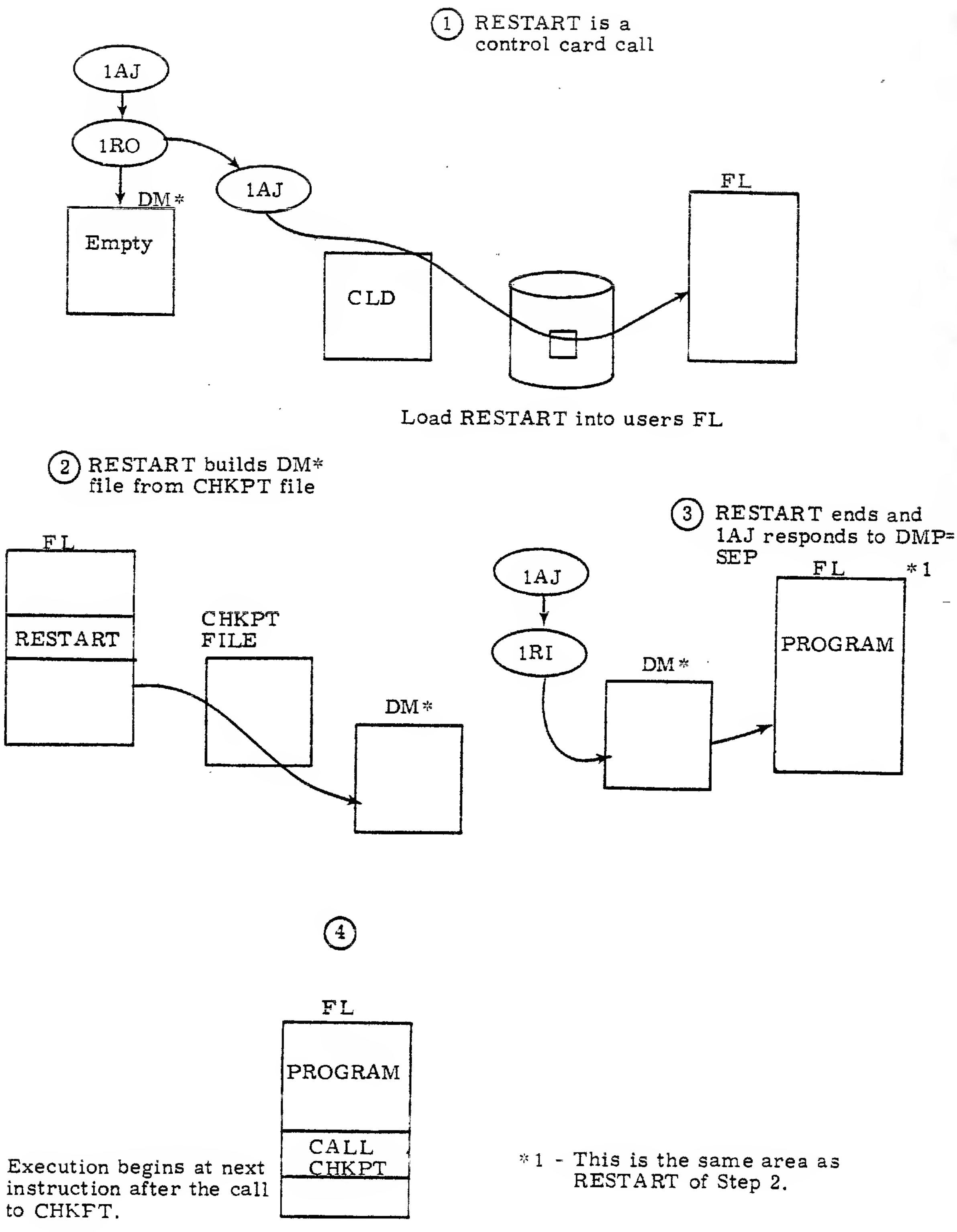


Figure 22-7. RESTART Overview

1AJ sets up CPA, SEPW, SPCW, etc., loads RESTART, stores the argument list in RA+ARGR, sets RA+ACTR accordingly, and initiates RESTART. RESTART cannot be called from an RA+1 request, so the parameter passing ability of DMP= is not utilized. RESTART locates the proper CKP file, requests the FL required, restores the files required (including the DM* file from the CKP file), and exits.

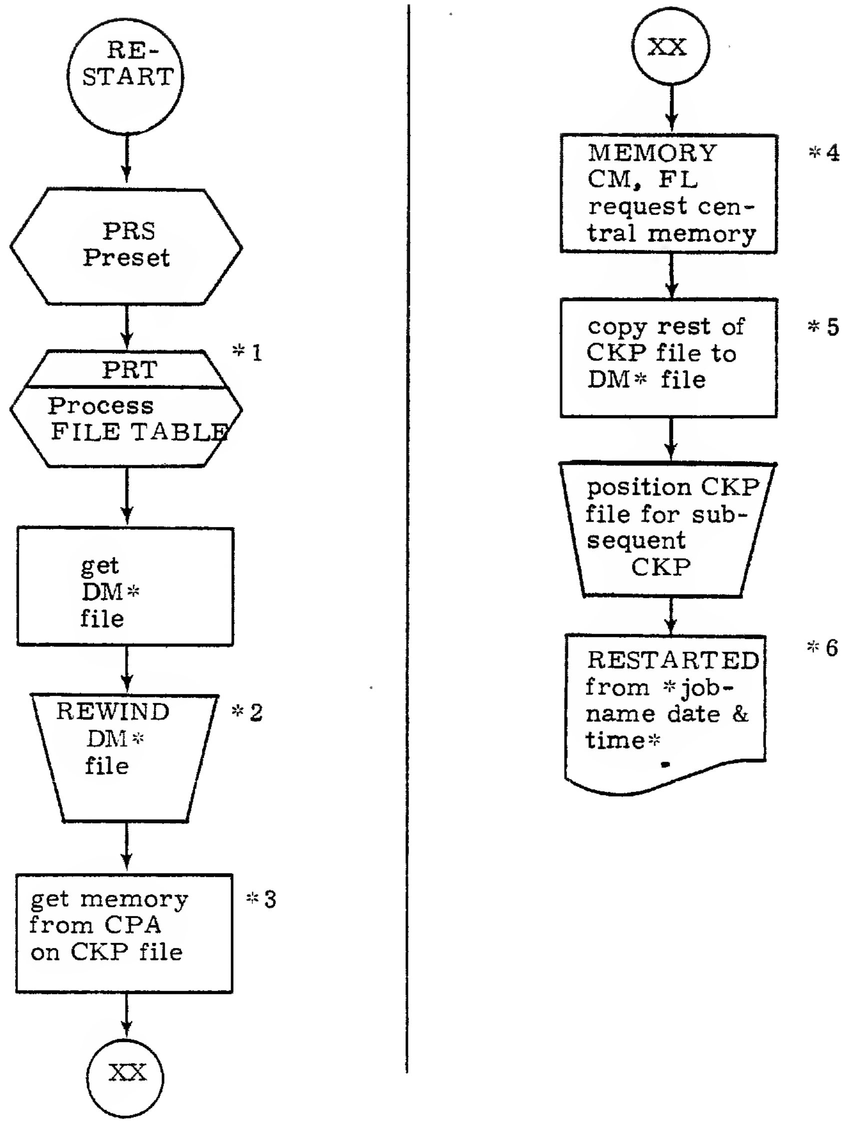
1AJ then finds the control point idle and notes this was a DMP= run. 1AJ will call 1RI, which rolls the job in using the DM* file created by RESTART. When 1RI is done, it clears the rollout flag, and the job is restarted from its position prior to checkpoint.

As in CHKPT, the preset routine is used as a buffer so that core is minimized. Table 22-2 lists some of the common decks used and the buffer assignments.

Figures 22-8 and 22-9 are flowcharts detailing the RESTART Main Loop and Preset Routine.

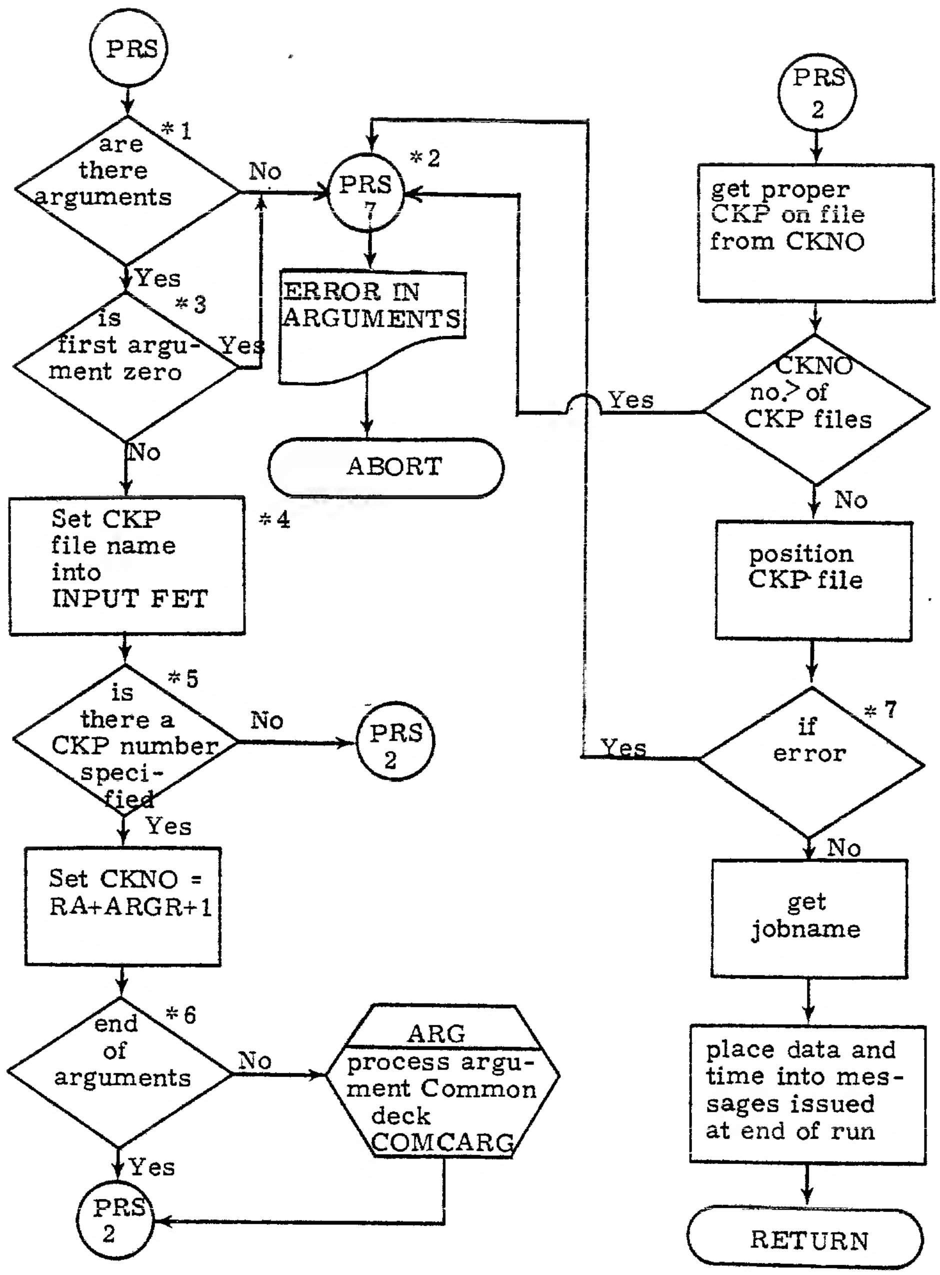
TABLE 22-2. RESTART BUFFER ASSIGNMENTS/COMMON DECKS

Load Address	Common Decks			
602		CTEXT	COMCARG	- Process arguments
626		CTEXT	COMCCDD	- Constant to decimal display code conversion
641		CTEXT	COMCCIO	- I/O function processor
655		CTEXT	COMCCPM	- Control point manager processor
661		CTEXT	COMCDXB	- Display code to binary conversion
676		CTEXT	COMCEDT	- Edit date or time from packed format
722		CTEXT	COMCLFM	- Local file manager processor
732		CTEXT	COMCPFM	- Permanent file processor
742		CTEXT	COMCRDC	- Read coded line, -C- format
754		CTEXT	COMCRDO	- Read one word
776		CTEXT	COMCROW	- Read words to working buffer
1112		CTEXT	COMCSFN	- Space fill name
1124		CTEXT	COMCSYS	- Process system request
1155		CTEXT	COMCWTO	- Write one word
1173		CTEXT	COMCWIW	- Write words from working buffer
		Buffer Assignments		
		USE	Buffers	
1300	BUF	EQU	*	
2301	IBUF	EQU	BUF+BUFL	
4302	OBUF	EQU	IBUF+IBUFL	
6303	SBUF	EQU	OBUF+OBUFL	
6704	TBUF	EQU	SBUF+SBUFL	
7707	RFL=	EQU	TBUF+TBUFL	



*1 Copy all data files from the CKP file.
2 Prepare to write remainder of CKP file onto the DM file.
*3 Exchange package in first 20B of CPA and word CPA+2 is FL.
*4 If job card FL request is less than FL needed to RESTART, control point would be aborted by CPUMTR.
*5 Stop on EOR.
*6 Message set up by preset.

Figure 22-8. RESTART - RESTART (Main Loop)



*1 Is (RA+ACTR) not equal zero.
*2 RESTART must have an argument list.
*3 Is (RA+ARGR)=0.
*4 (RA+ARGR)=CKP file name.
*5 Is (RA+ARGR+1)=0. Note that CKNO is preset to 0 at assembly.
*6 Is (RA+ARGR+2)=0.
*7 The error CHECKPOINT FILE error if header word missing or CHECKPOINT NOT FOUND if asked for a CKP number that is not on the file.

Figure 22-9. PRS - RESTART (Preset)

23.0 INTRODUCTION

When programming at the CP level, the system programmer has many debugging aids at his disposal. Among them are KCL, relative core dump macro and control card, DIS, and KRONREF. However, when programming at the PP level, the system programmer has fewer debugging aids (KCL, DIS, and deadstart dumps).

This section discusses the debugging aids available. At the end of this section is a listing of four useful debugging programs that are only available in this document.

23.1 KCL AND PROCEDURE FILES

KCL gives the user great versatility in coding a control card stream. The user can cause a job to execute in many different ways based on some selection criteria If a job sets a fatal error flag, the user may regain control by having an EXIT card in the control stream

KCL also has an extremely useful feature called procedure files. These can be used to good advantage while debugging a job, or even for some multi-task operations. The KCL descriptions and the procedure file description is given in Section 4 of the KRONOS 2 1 Reference Manual.

23.2 DOCMENT CARD

Internal and external documentation is contained in the listings of most of the programs on the system. By the use of the DOCMENT control, this documentation can be dumped to a printer. Page 31 of the KRONOS 2.1 Reference Manual describes how to use this card.

23.3 DIS AND QIS

Debugging also can be done by use of DIS. DIS gives the facility of breakpoint and 026.

These commands and features are discussed in the KRONOS 2.1 Operators Guide and in the KRONOS 2.1 Instant Manual.

QIS is a remote terminal DIS device. This hardware is not normally supported by KRONOS 2.1. However, software has been written to use this device. The software is not supported and not normally available. The CMR entry and binary EST entry for this device is:

EQxx = SC, St, eq, un, ch.

EST entry = 00pp 00cc 0000 2303 e0uu

where pp is CP number assigned to = 0.

cc is channel number

2303 is SC in display code

3 is equipment number

uu is unit number

23.4 DEADSTART DUMPS

The process to obtain deadstart dumps is detailed in Part II, Section 3 of the KRONOS 2.1 Installation Handbook. Part IV, Section 4 of the KRONOS 2.1 Installation Handbook details how to use the utility DSDI to format the express dump obtained using the deadstart F option.

23.5 FOUR USEFUL ROUTINES

The following four routines are described in the beginning of their code. The listings are shown in Figures 23-1 through 23-5.

23.5.1

Since the DUMP routines in KRONOS will not allow any CP user to dump absolute memory, the following routine was written (Figures 23-1 and 23-2). It consists of a CP routine which calls a PP routine. They will dump absolute memory as specified on the call card to the program. The user must SYSEDIT the PP program onto the system (Job 1) then run Job 2 The dump parameters are specified on the LGO card as follows. LGO (from field, to field). The dump will start at the from field and terminate one buffer beyond the to field . Illegal parameters are flagged in the listing. An example of some dumps are given at the end of the listings. Note a macro CKIR has been defined in TLP. This macro can be copied and used by any PP programmer who would like a way to unhang his PP during debugging. Used in the code at WAIT, the PP hangs while waiting for the CP program to respond. If the CP program cannot respond, the operator can cause the PP program to drop by changing the PP name in the IR in CMR from the console. When the PP has dropped, the CP program can be dropped by the drop command.

23.5.2

KRONOS 2 1 has no facility for register snap shots which will keep all the registers intact. The user can use the macro SYSTEM DMP, R, but X6, A6.A1, and X1 are destroyed and the macro makes a call to the system which has to load CPMEM. When control is returned to the user none of the registers can be assumed to be the same as before the call.

The following routine REGDMP (Figure 23-3) is loaded as a local routine and can be jumped to with out the system intervening All the registers are dumped intact to the printer and when control is returned to the caller, all the registers are intact. Hence, this routine can be called indiscriminantly for debugging purposes In addition it prints the address it was called from. The call is RJ = XREGDMP.

Also, entry points LOADEM and SAVEM can be used to save the registers and restore them at some later date. Note that a call to SAVEM will destroy the contents of the registers, but LOADEM will restore them, and can be used to restore them any number of times Any new call to SAVEM will reset LOADEM. The call is RJ = XSAVEM and RJ = XLOADEM.

JOB 1 will create a binary deck of REGDMP.

23.5 3

Figure 23-4 is an example of a CP routine using special entry points It also demonstrates the use of monitor function "RSB" which is used here to read the Equipment Status Table (EST).

23.5.4

Figure 23-5 a PP routine, will dump any track of any mass storage device to the printer and does not require an FNT/FST entry

TLP, CM50000, T7777.
ACCOUNT, ML0, .
COMPASS.
SYSEDIT.
7/8/9 multipunch EOR indicator

```
           IOENT   TLP,TLP                                                 TLP          1
           PERIPH                                                          TLP          2
           SST                                                             TLP          3
           XREF    A                                                       TLP          4
***        PP PORTION OF  ABSOMP  ROUTINE                                  NUMONE       1
*          CONTROL DATA CORPORATION                                        TLP          6
*          M. L. OMMERMAN                                                  TLP          7
*                                                                          TLP          9
*      ROUTINE   TO DUMP ABSOLUTE CORE WITH DISPLAY VARIABLES              TLP         10
*                USING  CP ROUTINE TALKPP  FOR OUTPUT BUFFER               TLP         11
*                                                                          TLP         12
**         PP   IR                                                         NUMONE       2
*T   IR  =  18/TLP,24/0,18/PPP                                             NUMONE       3
*       PP ROUTINE  IR = TLP0000PPP                                        TLP         13
*                WHERE   TLP IS PP NAME                                    TLP         14
*       PPP IS ADDRESS OF AUTO-RECALL WORD -PPONE- IN CP ROUTINE           TLP         15
*T   PPP =  24/FFF,18/0,18/TTT                                             NUMONE       4
*          PPONE  =   0FFF000TTT                                           TLP         16
*            WHERE  FFF  IS  18 BIT  FROM  FIELD                           TLP         17
*                        0    IS BIN ZERO                                  TLP         18
*                   TTT  IS  18 BIT  NUMBER OF BUFFERS TO DUMP             TLP         19
*                           NONZERO WHEN DONE -- NEXT WRD IS OUTPUT FET    TLP         20
OPL        XTEXT  COMPMAC                                                    NUMON  23405

 MKMSG     MACRO  *                                                        TLP         22
           LOC    MSG1                                                     TLP         23
           RJM    OFM                                                      TLP         24
           ENDM                                                            TLP         25
 CKIR      MACRO  *            CHECK TO SEE IF IR HAS CHANGED              TLP         26
           LOCAL  OK1                                                      TLP         27
           LDD    IA                                                       TLP         28
           CRD    IR           READ IR                                     TLP         29
           LDD    IR+1                                                     TLP         30
           SHN    -6           STIP OFF CP INFO                            TLP         31
           ADC    -1RP                                                     TLP         32
           ZJN    OK1                                                      TLP         33
           LDD    IR                                                       TLP         34
           ADC    -2RTL                                                    TLP         35
           ZJN    OK1                                                      TLP         36
           LJM    TLP14        STOP                                        NUMONE       6
 OK1       BSS    0                                                        TLP         38
           ENDM                                                            TLP         39
    20   CNT       EQU    20B                                              TLP         40
  7000   BUFFER    EQU    7000B                                            TLP         41
    40   BUFCNT    EQU    40B                                              TLP         42
    41   CMAN      EQU    41B                                              TLP         43
```

Figure 23-1. Job 1

```
                      42     CHAD      EQU    42B                                      TLP          44
1100                                   ORG    PPFW                                     TLP          45
1100                         TLP       BSS    0                                        TLP          46
1100        1400                       LDN    0                                        TLP          47
1101        3442                       STD    CHAD         START WITH CH ADDRESS 0     TLP          48
1102        3441                       STD    CHAN                                     TLP          49
1103        3050                       LDD    IR                                       TLP          50
1104        5400 1705                  STM    IRIMG                                    TLP          51
1106        2177 5363                  ADC    -2RTL        CHK FOR PROPER  CALL        TLP          52
1110        0510                       NJN    TLP1                                     NUMONE        7
1111        3051                       LDD    IR+1                                     TLP          54
1112        5400 1706                  STM    IRIMG+1                                  TLP          55
1114        1071                       SHN    -6           STRIP OFF CP INGO           TLP          56
1115        2177 7757                  ADC    -1RP                                     TLP          57
1117        0414                       ZJN    TLP3                                     NUMONE        8
1120        1411             TLP1      LDN    9                                        NUMONE        9
1121        3401                       STD    T1                                       TLP          60
1122        5001 1662        TLP2      LDM    ILCAL-1,T1   STORE ILCAL MSG INTO        NUMONE       10
1124        5401 1673                  STM    LCAL-1,T1                                TLP          62
1126        3701                       SOD    T1                                       TLP          63
1127        0572                       NJN    TLP2                                     NUMONE       11
1130        3051                       LDD    IR+1                                     TLP          65
1131        5400 1706                  STM    IRIMG+1                                  TLP          66
1133                         TLP3      BSS    0                                        NUMONE       12
1133        3052                       LDD    IR+2                                     TLP          68
1134        1071                       SHN    -6                                       TLP          69
1135        0200 2054                  RJM    C2D          CONVERT BIN TO DISPLAY CODE TLP          70
1137        5400 1707                  STM    IRIMG+2                                  TLP          71
1141        3052                       LDD    IR+2                                     TLP          72
1142        0200 2054                  RJM    C2D                                      TLP          73
1144        5400 1710                  STM    IRIMG+3                                  TLP          74
1146        3053                       LDD    IR+3                                     TLP          75
1147        1071                       SHN     -6                                      TLP          76
1150        0200 2054                  RJM    C2D                                      TLP          77
1152        5400 1711                  STM    IRIMG+4                                  TLP          78
1154        3053                       LDD    IR+3                                     TLP          79
1155        0200 2054                  RJM    C2D                                      TLP          80
1157        5400 1712                  STM    IRIMG+5                                  TLP          81
1161        3054                       LDD    IR+4                                     TLP          82
1162        1071                       SHN    -6                                       TLP          83
1163        0200 2054                  RJM    C2D                                      TLP          84
1165        5400 1713                  STM    IRIMG+6                                  TLP          85
1167        3054                       LDD    IR+4                                     TLP          86
1170        0200 2054                  RJM    C2D                                      TLP          87
1172        5400 1714                  STM    IRIMG+7                                  TLP          88
1174        5000 1674                  LDM    LCAL                                     TLP          89
1176        2177 6372                  ADC    -2RLE        CHK FOR LEGAL CALL          TLP          90
1200        0403                       ZJN    *+3                                      TLP          91
1201        0100 1636                  LJM    TLP14        STOP                        NUMONE       13
1203        3075                       CKIR   *            SEE IF SOMEONE  WANT TO STOP  EARLY   TLP   93
1220        3055                       LDD    RA                                       TLP          94
1221        1006                       SHN    6                                        TLP          95
1222        3154                       ADD    IR+4         GET  REST OF  F PPDONE FWA. TLP          96
1223        1006                       SHN    6                                        TLP          97
1224        3153                       ADD    IR+3                                     TLP          98
1225        1014                       SHN  12                                         TLP          99
1226        1601                       ADN    1            GET FWA OF FET              TLP         100
```

Figure 23-1. Job 1 (Cont'd)

```
1227   5400 2020         STH   FETLOW      STORE FET  FWA  LOWER 12 BITS           TLP      101
1231   1615              ADN   13          GET  BUF  ADD IN  FWA                   TLP      102
1232   5400 2021         STH   BUFCUR      STORE BUF  FWA  LOWER 12 BITS           TLP      103
1234   1715              SBN   13                                                  TLP      104
1235   1063              SHN   -12                                                 TLP      105
1236   5400 2017         STH   FETTOP      STORE FET AND BUF FWA  TOP 6 BITS       TLP      106
1240   5400 2024         STH   BUFTOP      TOP 6 BITS OF BUFFER FWA                TLP      107
1242   5000 2017         LDH   FETTOP                                              TLP      108
1244   1014              SHN   12                                                  TLP      109
1245   5100 2020         ADH   FETLOW                                              TLP      110
1247   1701              SBN   1                                                   TLP      111
1250   6001              CRD   T1          GET  PP DONE                            TLP      112
1251   3001              LDD   T1                                                  TLP      113
1252   3441              STD   CHAN        TOP 6 BITS OF  FROM  FIELD              TLP      114
1253   3002              LDD   T2                                                  TLP      115
1254   3442              STD   CHAD        LOW 12 BITS  OF  FROM  FIELD            TLP      116
1255   3005              LDD   T5          GET NO OF BUFFERS TO DUMP               TLP      117
1256   0702              MJN   TLP4                                                NUMONE    14
1257   0502              NJN   TLP5                                                NUMONE    15
1260   1401        TLP4  LDN   1           INSURE CNT IS ALWAYS GREATER  ZERO      NUMONE    16
1261               TLP5  BSS   0                                                   NUMONE    17
1261   3420              STD   CNT         NO OF BUFFERS  TO  DUMP                 TLP      122
1262   1472              LDN   72B         LENGTH  OF  BUFFER IN LINES             TLP      123
1263   3440              STD   BUFCNT      PP BUF CNT OF PP WORDS = 100B CP WORDS  TLP      124
1264               TLP6  BSS   0                                                   NUMONE    18
1264   3041              LDD   CHAN                                                TLP      126
1265   1277              LPN   77B         KEEP LOWER  6 BITS                      TLP      127
1266   1014              SHN   12                                                  TLP      128
1267   3142              ADD   CHAD                                                TLP      129
1270   6010              CRD   CH                                                  TLP      130
1271   3042              LDD   CHAD                                                TLP      131
1272   0200 2054         RJM   C2D         LOWER 6 BITS                            TLP      132
1274   5400 1721         STH   ADRS+1                                              TLP      133
1276   3042              LDD   CHAD                                                TLP      134
1277   1071              SHN   -6                                                  TLP      135
1300   0200 2054         RJM   C2D                                                 TLP      136
1302   5400 1720         STH   ADRS        PUT  ADDRESS INTO  ADRS  POSITION       TLP      137
1304   3041              LDD   CHAN                                                TLP      138
1305   0200 2054         RJM   C2D                                                 TLP      139
1307   5400 1717         STH   ADRT                                                TLP      140
                   *     GET 1ST  WORD                                             TLP      141
1311   1405              LDN   5                                                   TLP      142
1312   3401              STD   T1                                                  TLP      143
1313   1411              LDN   9                                                   TLP      144
1314   3402              STD   T2                                                  TLP      145
1315   5001 0007   TLP7  LDH   CH-1,T1                                             NUMONE    19
1317   0503              NJN   TLP8                                                NUMONE    20
1320   2055 5555         LDC   2H                                                  TLP      148
1322               TLP8  BSS   0                                                   NUMONE    21
1322   5401 1737         STH   D1-1,T1                                             TLP      150
1324   5001 0007         LDH   CH-1,T1                                             TLP      151
1326   0200 2054         RJM   C2D                                                 TLP      152
1330   5402 1724         STH   BIN1,T2                                             TLP      153
1332   3702              SOD   T2                                                  TLP      154
1333   5001 0007         LDH   CH-1,T1                                             TLP      155
1335   1071              SHN   -6          GET  TOP  TWO DIGITS                    TLP      156
1336   0200 2054         RJM   C2D                                                 TLP      157
```

Figure 23-1. Job 1 (Cont'd)

```
1340     5402 1724            STM    BIN1,T2                                    TLP      158
1342     3702                 SOD    T2                                         TLP      159
1343     3701                 SOD    T1                                         TLP      160
1344     0550                 NJN    TLP7                                       NUMONE    22
1345     3041                 LOD    CMAN                                       TLP      162
1346     1014                 SHN    12                                         TLP      163
1347     3142                 ADD    CMAD                                       TLP      164
1350     1601                 ADN    1        UPDATE CM ADDRESS                 TLP      165
1351     6010                 CRD    CM       READ 2ND PARCEL OF  CORE          TLP      166
1352     5400 0042            STM    CMAD                                       TLP      167
1354     1063                 SHN    -12                                        TLP      168
1355     5400 0041            STM    CMAN     UPDATE CM ADDRESS IN MEMORY       TLP      169
1357     1405                 LDN    5                                          TLP      170
1360     3401                 STD    T1                                         TLP      171
1361     1411                 LDN    9                                          TLP      172
1362     3402                 STD    T2                                         TLP      173
1363     5001 0007    TLP9    LDM    CM-1,T1                                    NUMONE    23
1365     0503                 NJN    TLP10                                      NUMONE    24
1366     2055 5555            LDC    2H                                         TLP      176
1370                  TLP10   BSS    0                                          NUMONE    25
1370     5401 1762            STM    D2-1,T1                                    TLP      178
1372     5001 0007            LDM    CM-1,T1                                    TLP      179
1374     0200 2054            RJM    C2D                                        TLP      180
1376     5402 1747            STM    BIN2,T2                                    TLP      181
1400     3702                 SOD    T2                                         TLP      182
1401     5001 0007            LDM    CM-1,T1                                    TLP      183
1403     1071                 SHN    -6                                         TLP      184
1404     0200 2054            RJM    C2D                                        TLP      185
1406     5402 1747            STM    BIN2,T2                                    TLP      186
1410     3702                 SOD    T2                                         TLP      187
1411     3701                 SOD    T1                                         TLP      188
1412     0550                 NJN    TLP9                                       NUMONE    26
1413     3041                 LOD    CMAN                                       TLP      190
1414     1014                 SHN    12                                         TLP      191
1415     3142                 ADD    CMAD                                       TLP      192
1416     1601                 ADN    1        UPDATE CM ADDRESS                 TLP      193
1417     6010                 CRD    CM                                         TLP      194
1420     5400 0042            STM    CMAD                                       TLP      195
1422     1063                 SHN    -12                                        TLP      196
1423     5400 0041            STM    CMAN     UPDATE CM ADDRESS IN MEMORY       TLP      197
1425     1405                 LDN    5                                          TLP      198
1426     3401                 STD    T1                                         TLP      199
1427     1411                 LDN    9                                          TLP      200
1430     3402                 STD    T2                                         TLP      201
1431     5001 0007    TLP11   LDM    CM-1,T1                                    NUMONE    27
1433     0503                 NJN    TLP12                                      NUMONE    28
1434     2055 5555            LDC    2M                                         TLP      204
1436                  TLP12   BSS    0                                          NUMONE    29
1436     5401 2005            STM    D3-1,T1                                    TLP      206
1440     5001 0007            LDM    CM-1,T1                                    TLP      207
1442     0200 2054            RJM    C2D                                        TLP      208
1444     5402 1772            STM    BIN3,T2                                    TLP      209
1446     3702                 SOD    T2                                         TLP      210
1447     5001 0007            LDM    CM-1,T1                                    TLP      211
1451     1071                 SHN    -6                                         TLP      212
1452     0200 2054            RJM    C2D                                        TLP      213
1454     5402 1772            STM    BIN3,T2                                    TLP      214
```

Figure 23-1. Job 1 (Cont'd)

```
1456        3702               SOD   T2                                           TLP       215
1457        3701               SOD   T1                                           TLP       216
1460        0550               NJN   TLP11                                        NUMDNE     30
1461        3041               LDD   CMAN                                         TLP       218
1462        1014               SHN   12                                           TLP       219
1463        3142               ADD   CMAD                                         TLP       220
1464        1601               ADN   1          UPDATE CM ADDRESS                 TLP       221
1465        5400 0042          STM   CMAD                                         TLP       222
1467        1063               SHN   -12                                          TLP       223
1470        5400 0041          STM   CMAN       UPDATE CM ADDRESS IN MEMORY       TLP       224
1472        1415               LDN   13                                           TLP       225
1473        5400 0001          STM   T1         NO  OF  WORDS TO  TRANSFER        TLP       226
1475        5000 2024          LDM   BUFTOP                                       TLP       227
1477        1014               SHN   12                                           TLP       228
1500        5100 2021          ADM   BUFCUR                                       TLP       229
1502        6301 1716          CWM   NEXTLN,T1  TRANSFER  12 WORDS TO CM          TLP       230
1504        5400 2021          STM   BUFCUR     CWM WILL INCREMTNT (A) TO (A+D)   TLP       231
1506        1063               SHN   -12                                          TLP       232
1507        5400 2024          STM   BUFTOP                                       TLP       233
1511        3075               CKIR  *          SEE IF SOMEONE  WANT TO STOP  EARLY  TLP    234
1526        3740               SOD   BUFCNT                                       TLP       235
1527        0403               ZJN   *+3                                          TLP       236
1530        0100 1264          LJM   TLP6       GET NEXT 3 WORDS OF CORE          NUMDNE     31

1532        1400               LDN   0                                            TLP       239
1533        3401               STD   T1         SET UP CM WORD PPDONE             TLP       240
1534        3402               STD   T2         SET UP CM WORD PPDONE             TLP       241
1535        3403               STD   T3         SET UP CM WORD PPDONE             TLP       242
1536        3404               STD   T4         SET UP CM WORD PPDONE             TLP       243
1537        1401               LDN   1                                            TLP       244
1540        3405               STD   T5         SET UP CM WORD PPDONE             TLP       245
1541        5000 2017          LDM   FETTOP                                       TLP       246
1543        1014               SHN   12                                           TLP       247
1544        5100 2020          ADM   FETLOW                                       TLP       248
1546        1701               SBN   1                                            TLP       249
1547        6201               CWD   T1                                           TLP       250
1550        1601               ADN   1                                            TLP       251
1551        6001               CRD   T1         GET FET WORD                      TLP       252
1552        1400               LDN   0                                            TLP       253
1553        3404               STD   T4                                           TLP       254
1554        3405               STD   T5                                           TLP       255
1555        3605               ADD   T5                                           TLP       256
1556        5000 2017          LDM   FETTOP                                       TLP       257
1560        1014               SHN   12                                           TLP       258
1561        5100 2020          ADM   FETLOW                                       TLP       259
1563        6201               CWD   T1         INDICATE   PP ROUTINE DONE        TLP       260
1564        1472               LDN   72B        LENGTH  OF  BUFFER IN LINES       TLP       261
1565        3440               STD   BUFCNT     START BUFCNT  UP AGAIN            TLP       262
1566        3720               SOD   CNT                                          TLP       263
1567        0503               NJN   *+3                                          TLP       264
1570        0100 1636          LJM   TLP14                                        NUMDNE     32
1572        5000 2017          LDM   FETTOP                                       TLP       266
1574        5400 2024          STM   BUFTOP                                       TLP       267
1576        5000 2020          LDM   FETLOW                                       TLP       268
1600        1615               ADN   13         LEAVE  HEADER                     TLP       269
1601        5400 2021          STM   BUFCUR     RESTART BUF POINTER               TLP       270
```

Figure 23-1. Job 1 (Cont'd)

```
1603      1447               MONITOR  RCPM      REQUEST  CPU (Necessary to start CPU)        NUMONE     33
1606                  TLP13  BSS      0                                                     NUMONE     34
1606      3075               CKIR     *         CHECK IR                                    TLP       274
1623      5000 2017          LDM      FETTOP                                                TLP       275
1625      1014               SHN      12                                                    TLP       276
1626      5100 2020          ADM      FETLOW                                                TLP       277
1630      1701               SBN      1                                                     TLP       278
1631      6001               CRD      T1        READ  PPDONE                                TLP       279
1632      3005               LDD      T5                                                    TLP       280
1633      0552               NJN      TLP13                                                 NUMONE     35
1634      0100 1264          LJM      TLP6                                                  NUMONE     36
1636                  TLP14  BSS      0         SAME AS  STOP                               NUMONE     37
1636      2077 7776          LDC      -1                                                    TLP       284
1640      5400 0001          STM      T1                                                    TLP       285
1642      5000 2017          LDM      FETTOP                                                TLP       286
1644      1014               SHN      12                                                    TLP       287
1645      5100 2020          ADM      FETLOW                                                TLP       288
1647      2177 7776          ADC      -1                                                    TLP       289
1651      6201               CWD      T1        INDICATE ALL DONE                           TLP       290
1652      2000 2037          LDC      FINAL                                                 TLP       291
1654      0200 0501          RJM      DFM                                                   TLP       292
1656      1444               MONITOR  DPPM      DROP  PP (Necessary before ending)          NUMONE     38
1661      0100 0103          LJM      PPR                                                   TLP       295
1663      1114        ILCAL  DATA     18HILLEGAL TLP CALL =                                 TLP       296
1674      1405        LCAL   DATA     18HLEGAL TLP CALL =                                   TLP       297
1705              11  IRIMG  BSSZ     9                                                     TLP       298
1716      5555        NEXTLN DATA     2H                                                    TLP       299
1717      0000        ADRT   DATA     0                                                     TLP       300
1720               2  ADRS   BSSZ     2         ADDRESS OF CORE                             TLP       301
1722      5555               DATA     4H                                                    TLP       302
1724              12  BIN1   BSSZ     10        OCTAL REP OF  1ST WORD                      TLP       303
1736      5555               DATA     4H                                                    TLP       304
1740               5  D1     BSSZ     5         DISPLAY   OF  1ST WORD                      TLP       305
1745      5555               DATA     4H                                                    TLP       306
1747              12  BIN2   BSSZ     10                                                    TLP       307
1761      5555               DATA     4H                                                    TLP       308
1763               5  D2     BSSZ     5                                                     TLP       309
1770      5555               DATA     4H                                                    TLP       310
1772              12  BIN3   BSSZ     10        OCTAL REP OF  3RD WORD                      TLP       311
2004      5555               DATA     4H                                                    TLP       312
2006              11  D3     BSSZ     9         DISPLAY REP OF 3RD WORD + 12 BITS OF ZERO   TLP       313
2017      0000        FETTOP DATA     0         FET ADDRESS TOP 6  BITS AND BUF             TLP       314
2020      0000        FETLOW DATA     0         FET ADDRESS BOTTOM  12 BITS                 TLP       315
2021      0000        BUFCUR DATA     0         BUF FWA LOWER  12  BITS                     TLP       316
2022               2  DMPSZ  BSSZ     2         SIZE OF CM TO DUMP                          TLP       317
2024      0000        BUFTOP DATA     0         TOP 6 BITS OF CURRENT BUFFER                TLP       318
2025      2414        MSG1   DIS      ,*TLP CHECKING IR*                                    TLP       319
2036               1         BSSZ     1                                                     TLP       320
2037      0102        FINAL  DIS      ,*ABSOLUTE DUMP COMPLETE*                             TLP       321


                 D_M         BASE     M                                                     NUMONE     39
2053                  DPL    XTEXT    COMPC20   CONVERT 2 BIN DIGITS  TO DISPLAY CODE (C20)  NUMON  23405
2065                         END                                                            TLP       342
```

Figure 23-1. Job 1 (Cont'd)

```
          OMPDAGQ.   73/11/26. ASD KRONOS V2.1  SN 123  LEV=2  PSR 357
14.55.28.OMPDX,CM50000,T7777.
14.55.28.ACCOUNT,MLO,.
14.55.28.COMPASS.
14.55.33. ASSEMBLY COMPLETE.   44100B SCM USED.
14.55.33.    2.012  CPU SECONDS  ASSEMBLY TIME.
14.55.34.COPYBR(INPUT,LGO)
14.55.34.COPY COMPLETE.
14.55.34.LGO.
14.55.34.LGO(15)
14.55.35.LGO(1,)
14.55.36.LGO(,25)
14.55.36.LGO(0,200)
14.55.37.ABSOLUTE DUMP COMPLETE
14.55.37.LGO(0,1000B)
14.55.38.ABSOLUTE DUMP COMPLETE
14.55.39.LGO(478,369)
14.55.39.LGO(11000B,13000B)
14.55.40.ABSOLUTE DUMP COMPLETE
14.55.41.LGO(32000,35000)
14.55.43.ABSOLUTE DUMP COMPLETE
14.55.43.LGO(3,500)
14.55.44.ABSOLUTE DUMP COMPLETE
14.55.45.LGO(0,800)
14.55.45.ABSOLUTE DUMP COMPLETE
14.55.46.CP        3.277 SEC.
14.55.46.CM        0.017 KWH.
14.55.46.MS        0.828 KPR.
15.01.55.LP        2.739 KLN.
```

Figure 23-2. Job 2

```
DMPABS     ABSOUTE OUMP ROUTINE CP PORTION                    COMPASS 3.73309.     74/03/08. 21.47.20.      PAGE      2

                                  IOENT  ABSDMP                                                       AO5OMP         1
                                  ENTRY  ABSOMP                                                       MUMOME         1

                           ***    ABSOMP                                                              MUMOME         2
                           *      COPYRIGHT  COMTROL OATA CORPORATION                                 ABSOMP         5
                           *      AUTMOR     M. L. OHMERMAN                                           ABSOMP         6
                           *      OATE       OCT, 1973                                                ABSOMP         7

      0                     OPL   XTEXT  COMCOX8                                                        MUMOM  23405
     14                     OPL   XTEXT  COMCCOO                                                        NUMOM  23405
     24                     OPL   XTEXT  COMCCOD                                                        MUMOM  23405
     36                     OPL   XTEXT  COMCWOD                                                        NUMOM  23405

                           *      OUMP   ABSOLUTE  MEMORY                                             ABSDMP        12
                           *      CALLS  PP ROUTINE  TLP  (SUPPLIEO)                                  ABSOMP        13
                           *      THE CALL CONTROL CARO CONTAINS THE OUMP PARAMETERS                  ABSOMP        14
                           *       LGO(FROM,TO)                                                       ABSOMP        15
                           *         WHERE                                                            ABSOMP        16
                           *        FROM  IS  THE  START  OF CORE                                     ABSOMP        17
                           *         TO    IS  THE  LAST  OF  CORE  TO OUMP                           ABSDMP        18
                           *       LGO(100,25000) WILL OUMP ABSOLUTE CORE                             ABSOMP        19
                           *                   FROM  100   TO   25000                                 ABSOMP        20
                           *    OUMP PARAMETERS ARE PLACEO INTO PPOOME EITMER FIELO MAY BE            ABSOMP        21
                           *                      OCTAL WITH RADIX  B                                 ABSOMP        22

     54  00000000000000000000   SAVE    OATA   0                                                      ABSOMP        25
                            0   RA      EQU    0                                                      ABSOMP        26
     55  55051520243155555555   ZR      OATA   10H EMPTY                                              ABSOMP        27
     56  00000000000000000001   PAGEX   OATA   1                                                      ABSOMP        28
                           *       MOTE---  PPOONE MUST BE FOLLOWED BY OUTPUT FET                     ABSOMP        29
     57  00000000000000000000   PPOOME  OATA   0                                                      ABSOMP        30
     60  17252420252400000001   OUTPUT  VFO    36/6LOUTPUT,24/1     COMPLETE BIT MUST BE SET          ABSOMP        31
     61  00000000000000000065 +         VFO    60/BUF                                                 ABSDMP        32
     62  00000000000000000075 +         VFO    60/BUF+8                                               ABSOMP        33
     63  00000000000000000065 +         VFO    60/BUF                                                 ABSOMP        34
     64  00000000000000001466 +         VFO    60/BUF+1401B                                           AOSOMP        35
     65  34010223171425240555   BUF     OATA   20H1ABSOLUTE DUMP FROM                                 ABSDMP        36
     67  00000000000000000000   FROM    OATA   0                                                      ABSDMP        37
     70  55241755555555555555           OATA   10H TO                                                 ABSONP        38
     71  00000000000000000000   TO      DATA   0                                                      ABSDMP        39
     72  55555520010705555555           OATA   10H   PAGE                                             AOSDMP        40
     73  55555555553400000000   PAGE    VFO    36/6H     1,24/0                                       ABSDMP        41
     74  00000000000000000000           OATA   0                                                      ABSDMP        42
     75  55000000000000000000           VFO    6/1H ,54/0   SKIP OME LINE                             ABSOMP        43
     76                  1360           BSSZ   1360B                                                  ABSDMP        44
   1456  55555555555555555555           OATA   40H                                                    ABSOMP        45
   1462  55555555555555555555           OATA   20H                                                    ABSOMP        46
   1464                     2           BSSZ   2          EMO OF BUFFER                               AOSOMP        47
   1466                         ABSOMP  BSS    0                                                      NUMOME         8
   1466  7120000060 +                   OPEN   OUTPUT,WRITE,R                                         ABSOMP        51
   1470  5110000002                     SA1    RA+2       GET  FROM FIELD                             ABSOMP        52
```

Figure 23-2. Job 2 (Cont'd)

```
             0301001550 +        ZR      X1,ABSDMP4                              NUMONE     9
1471  0109001563 +               RJ      ABSDMP7                                 NUMONE    10
1472  5160000067 +               SA6     FROM                                    ABSOMP    55
             5110000003          SA1     RA+1       GET  TO  FIELO               ABSOMP    56
1473  0301301552 +               ZR      X1,ABSDMP5                              NUMONE    11
             0100001563 +        RJ      ABSDMP7                                 NUMONE    12
1474  5160000071 +               SA6     TO                                      ABSDMP    59
             7120000060 +        WRITER  OUTPUT,R       PUT OUT START MSG        ABSOMP    60
1476  5110000002                 SA1     RA+2       FROM FIELD                   NUMONE    13
             6170000001          SB7     1                                       ABSDMP    62
1477  6110000001                 SB1     1                                       ABSOMP    63
             10511               BX5     X1                                      ABSOMP    64
1500  0100000010 +               RJ      OXB        CONVERT FROM FIFLO TO INTEGER ABSDMP   65
1501  0314001554 +               NZ      X4,ABSOMP6                              NUMONE    14
             5160000057 +        SA6     PPOONE                                  ABSOMP    67
1502  10166                      BX1     X6                                      ABSOMP    68
         5100555555              SA0     555555B                                 AOSOMP    69
1503  0100001605 +               RJ      ABSDMP10                                NUMONE    15
1504  5160000067 +               SA6      FROM      USE OCTAL VALUE              ABSDMP    71
             5110000003          SA1     RA+3       TO FIELD                     NUMONE    16
1505  6170000001                 SB7     1                                       ABSDMP    73
             10511               BX5     X1                                      ABSOMP    74
1506  0100000010 +               RJ      OXB        CONVERT TO FIELO TO INTEGER  ABSOMP    75
1507  0314001554 +               NZ      X4,ABSOMP6                              NUMONE    17
             5110000057 +        SA1     PPOONE                                  ABSOMP    77
1510  5160000054 +               SA6     SAVE                                    ABSDMP    78
             10166               BX1     X6                                      ABSOMP    79
1511  5100555555                 SA0     555555B                                 ABSOMP    80
             0100001605 +        RJ      AOSOMP10                                NUMONE    18
1512  5160000071 +               SA6     TO         USE OCTAL VALUE              ABSOMP    82
             5110000057 +        SA1      PPOONE    FROM FIELD                   ABSOMP    83
1513  5120000054 +               SA2     SAVE       TO FIELD                     ABSOMP    84
             63210               SB2     X1                                      ABSDMP    85
                  67202          SB2     -B2                                     ABSDMP    86
1514  73322                      SX3     X2+B2      TO - FROM                    ABSOMP    87
         0333001554 +            NG      X3,ABSOMP6                              NUMONE    19
1515  0303001554 +               ZR      X3,ABSDMP6                              NUMONE    20
             27303               PX3     X3                                      ABSOMP    90
                  24303          NX3     X3                                      ABSOMP    91
1516  7150000253                 SX5     253B                                    ABSOMP    92
             27505               PX5     X5                                      ABSDMP    93
                  24505          NX5     X5                                      ABSOMP    94
1517  44435                      FX4     X3/X5                                   ABSOMP    95
         26424                   UX4     X4,B2                                   ABSOMP    96
             22424               LX4     X4,B2                                   ABSOMP    97
                  73441          SX4     X4+B1       GET NEXT LARGEST BUFFER NUMBER ABSDMP 98
1520  20144                      LX1     60-24      ONIT                         NUMONE    21
         12614                   BX6     X1+X4                                   ABSOMP   100
             5160000057 +        SA6      PPOONF                                 ABSOMP   101
1521  7160001465 +               SX6     OUF+14000                               ABSOMP   102
             5160000062 +        SA6     OUTPUT+2    PREPATR FOR FIRST WRITE     ABSOMP   103
1522  7160000065 +               SX6     BUF                                     ABSOMP   104
             5066000001          SA6     A6+1                                    ABSOMP   105
1523  7160241420                 SYSTEM  TLP,R,PPOONE                            NUMONF    22
1526                   ABSOMP1    BSS    0                                       NUMONE    23
1526  5110000057 +               SA1     PPOONE     SEE IF  TLP  OONE            ABSOMP   114
             0301001544 +        ZR      X1,ABSOMP3                              NUMONE    24
```

Figure 23-2. Job 2 (Cont'd)

```
JMPABS     ABSOUTE DUMP ROUTINE CP PORTION                    COMPASS 3.73309.     74/03/08. 21.47.20.      PAGE      4

  1527  0331001540 +                    NG     X1,ABSDMP2                                                    NUMONE      25
                7120000060 +            WRITE  OUTPUT,R                                                      ABSDMP     117
  1531  7160001465 +                    SX6    BUF+1400B                                                     ABSDMP     118
                5160000062 +            SA6    OUTPUT+2     PREPAIR  FOR  NEXT  WRITE                        ABSDMP     119
  1532  5110000056 +                    SA1    PAGEX                                                         ABSDMP     120
                7160000065 +            SX6    BUF                                                           ABSDMP     121
  1533  5066000001                      SA6    A6+1                                                          ABSDMP     122
                7261000001              SX6    X1+1                                                          ABSDMP     123
  1534  5160000056 +                    SA6    PAGEX                                                         ABSDMP     124
                73160                   SX1    X6                                                            ABSDMP     125
                     56000              SA0    B0           PROPIGATE BIN ZEROES                             ABSDMP     126
  1535  6110000001                      SB1    1            ADDR    REQUIRES B1=1                            ABSDMP     127
                0100000027 +            RJ     CDD          CONVERT PAGE NO TO BCD                           NUMONE      26
  1536  5160000073 +                    SA6    PAGE                                                          ABSDMP     129
                76600                   SX6    B0                                                            ABSDMP     130
  1537  5160000057 +                    SA6    PPDONE       TELL PP  READY  TO  GO                           ABSDMP     131
                0400001544 +            EQ     ABSDMP3                                                       NUMONE      27
  1540                        ABSDMP2   BSS    0                                                             NUMONE      28
  1540  7120000060 +                    WRITEF   OUTPUT,R                                                    ABSDMP     134
  1542  7160247021                      ENDRUN           TLP  DONE                                           ABSDMP     135
  1544                        ABSDMP3   BSS    0                                                             NUMONE      29
  1544  7160220314                      SYSTEM RCL,R,PPDONE                                                  NUMONE      30
  1547  0400001526 +                    EQ     ABSDMP1                                                       NUMONE      31
  1550  5110000055 +          ABSDMP4   SA1    ZR                                                            NUMONE      32
                10611                   BX6    X1                                                            ABSDMP     144
  1551  5160000067 +                    SA6    FROM                                                          ABSDMP     145
  1552  5110000055 +          ABSDMP5   SA1    ZR                                                            NUMONE      33
                10611                   BX6    X1                                                            ABSDMP     147
  1553  5160000071 +                    SA6    TO                                                            ABSDMP     148
  1554                        ABSDMP6   BSS    0                                                             NUMONE      34
  1554  5110001561 +                    SA1    ER                                                            ABSDMP     150
                5021000001              SA2    A1+1                                                          ABSDMP     151
  1555  10611                           BX6    X1                                                            ABSDMP     152
           10722                        BX7    X2                                                            ABSDMP     153
                5160000072 +            SA6    TO+1                                                          ABSDMP     154
  1556  5076000001                      SA7    A6+1         SET ERROR MESSAGE                                ABSDMP     155
                7160000075 +            SX6    BUF+8                                                         ABSDMP     156
  1557  5160000062 +                    SA6    OUTPUT+2       WRITE ERROR MESSAGE                            ABSDMP     157
                7170000065 +            SX7    BUF                                                           ABSDMP     158
  1560  5076000001                     SA7   A6+1           RESTORE IN POINTER                               ABSDMP     159
                0400001540 +            EQ     ABSDMP2                                                       NUMONE      35
  1561  55012207251505162455    ER      DATA   20H ARGUMENT ERROR                                            ABSDMP     161
```

Figure 23-2. Job 2 (Cont'd)

```
CDNVERT CONTROL CARD ARGUMENT TD DXB TYPE INPUT

  1563  00000000000000000000        ABSDMP7  DATA    0                                                      NUMDNE       36
                                    *         ENTRY   X1= LT JUSTIFIED DPC TERMINATED BY 6 BITS OF ZERO    ABSDMP      245
                                    *         EXIT    X6= RT JUSTIFIED DPC CONSTANT                        ABSDMP      246
                                    *          DEMAND  B1 = 1                                              ABSDMP      247
                                    *         USES  B1,B6,X0,X1,X2,X6,A2                                   ABSDMP      248
                                    *                                                                      ABSDMP      249
  1564  7100000077                           SX0     77B        DPC  GETTER                                ABSDMP      250
                  43600                      MX6     0                                                     ABSDMP      251
  1565  6160000012                           SB6     10                                                    ABSDMP      252
  1566  20106                       ABSDMP8  LX1     6                                                     NUMDNE       37
             11210                           BX2     X1*X0      GET  TDP DPC                               ABSDMP      254
                  0302001571 +               ZR      X2,ABSDMP9                                            NUMDNE       38
  1567  20606                                LX6     6                                                     ABSDMP      256
             12662                           BX6     X6+X2                                                 ABSDMP      257
                  67661                      SB6     B6-B1                                                 ABSDMP      258
  1570  0661001566 +                         GE      B6,B1,ABSDMP8                                         NUMDNE       39
                  0460001563 +               EQ      B6,ABSDMP7                                            NUMDNE       40
  1571                              ABSDMP9  BSS     0                                                     NUMDNE       41
  1571  5126001572 +                         SA2     SPCTAB+B6  GET PRDPER ND DF SPACES                    ABSDMP      262
                  12662                      BX6     X6+X2                                                 ABSDMP      263
  1572  0400001563 +                SPCTAB   EQ      ABSDMP7    SPCTAB+B6  WHERE B6=0  IS  IMPDSSIBLE    NUMDNE       42
  1573  55000000000000000000        +        VFD     6/1H ,54/0                                            ABSDMP      265
  1574  55550000000000000000                 VFD     12/2H  ,48/0                                          ABSDMP      266
  1575  55555500000000000000                 VFD     18/3H   ,42/0                                         ABSDMP      267
  1576  55555555000000000000                 VFD     24/4H    ,36/0                                        ABSDMP      268
  1577  55555555550000000000                 VFD     30/5H     ,30/0                                       ABSDMP      269
  1600  55555555555500000000                 VFD     36/6H      ,24/0                                      ABSDMP      270
  1601  55555555555555000000                 VFD     42/7H       ,18/0                                     ABSDMP      271
  1602  55555555555555550000                 VFD     48/8H        ,12/0                                    ABSDMP      272
  1603  55555555555555555500                 VFD     54/9H         ,6/0                                    ABSDMP      273
  1604  00000000000000000000                 VFD     60/0       ILLEGAL                                    ABSDMP      274
```

Figure 23-2. Job 2 (Cont'd)

```
DMPABS      ABSOUTE DUMP ROUTINE CP PORTION                    COMPASS 3.73309.     74/03/08. 21.47.20.      PAGE     6
ADDR CONVERT 18 BIT ADDRESS TO OCTAL

                                  *          ENTRY     X1 = BINARY ADDRESS IN LOWER 18 BITS                 ABSDMP     276
                                  *          EXIT      X6 = OCTAL CONVERSION IN TOP 36 BITS                 ABSDMP     277
                                  *                 LOWER 24 BITS PROPIGATED WITH VALUE OF A0               ABSDMP     278
                                  *          USES   B1,B2,B3,B6,X0,X1,X2,X6,A0                              ABSDMP     279
                                  *          ALSO USES X3                                                   ABSDMP     280
                                  *              DEMAND  THAT  B1 = 1                                       ABSDMP     281
  1605                    1     ABSDMP10  BSS   1                                                           NUMONE      43
  1606  0100000036 +                      RJ    WOD         CONVERT TO OCTAL                                NUMONE      44
  1607  43030                             MX0   24                                                          NUMONE      45
            15770                         BX7    -X0*X7                                                     NUMONE      46
                20730                     LX7   24                                                          NUMONE      47
                    74200                 SX2   A0                                                          NUMONE      48
  1610  43052                             MX0   60-18                                                       NUMONE      49
            15220                         BX2   -X0*X2                                                      NUMONE      50
                7100000077                SX0   77B                                                         NUMONE      51
  1611  11020                             BX0   X2*X0       GET ONE EXTRA CHAR                              NUMONE      52
            20206                         LX2   6                                                           NUMONE      53
                12220                     BX2   X2+X0                                                       NUMONE      54
                    12672                 BX6   X7+X2                                                       NUMONE      55
  1612  0400001605 +                      EQ    ABSDMP10                                                    NUMONE      56
  1613                                    END   ABSDMP                                                      NUMONE      57

                        42200B CM  STORAGE USED            410 STATEMENTS      42 SYMBOLS     000001 INVENTED SYMBOLS
                                   MODEL 73 ASSEMBLY      3.717 SECONDS       100 REFERENCES
```

Figure 23-2. Job 2 (Cont'd)

```
000014   01420302034303450447   A7CBCBC.D.   04510453045704630500   D(D$D.D:E   05020503050405110513   EBECEDEIEK
000017   05140515175700000000   ELEMO.       00000000000000000354          C=   23312324051555550000   SYSTEM
000022   00000000000101072200         AAGR   00000000000000001433          L0   00000101000000100010      HH  H H
000025   00000000000000000000                00000000004236363633       73330   00000000031332166745       CKZNA+
000030   55343757404057364257    14.55.37.   55423650343450354157   73/11/26.   55012304551322171617    ASD KRONO
000033   23552635573455552316   S V2.1  SN   55343536555514052646   123  LEV-   35555520232255364042   2  PSR 357
000036   55000000000000000000                00000000000000000000               00000000000000000001            A
000041   00000000541000000537        E6 E4   00000000000000020003         B C   00011200000000030000    AJ     C
000044   00000000000000260025          V U   00000000000000000000               00034131000000276200    C6Y   W]
000047   00000000000000006240           15   00000000000007601501        GEMA   00000000000000000000
000052   00000000000000000000                00000000000000000000               00000000000000000000
000055   00000000000000000000                00000000000000000000               77777776040004000000   |||-D D
000060   00000000060000000600       F    F   77770000000000000000   ||          00000000000000020043          B #
000063   00016736000000000000   A^3          61701073010010737773   [+H>A H>|>  00002003371061131073     PC4H[KH>
000066   00000000000000000000                00000000000000000000               00000000000000000000
000071   00000000000000000000                00000000000000000000               00000000000000000000
000074   00000000000000000000                00000000000000000000               04000000760000000000   D    ¬
000077   04000000770000000000   D    |       00000000000000000000               00000000000000010000            A
000102   00060000000000000000   F            01000000000000000000   A           00000000000000000000
000105   00000000000001000100          A A   00000536000001254235   E3  AU72    00000000000000000000
000110   00000000000000010001          A A   00000000055600000000      E,       00000000000000000000
000113   00000000000000000000                00000000000000000000               00000000000000000000
000116   00000000000000000000                00000000000000000000               00000000000000000000
000121   00000000000000000000                00000000000000000000               00000000000000000000
000124   00000000000000000000                00000000000000000000               00000000000000000000
000127   00000000000000000000                00000000000000000000               00000000000000000000
000132   00000000000000000000                00000000000000000000               00000000000000000000
000135   00000000000000000000                00000000000000000000               00000000000000000000
000140   00000000000000000000                00000000000000000000               00000000000000000000
000143   00000000000000000000                00000000000000000000               00000000000000000000
000146   00000000000000000000                00000000000000000000               00000000000000000000
000151   00000000000000000000                00000000000000000000               00000000000000000000
000154   00000000000000000000                00000000000000000000               00000000000000000000
000157   00000000000000000000                00000000000000000000               00000000000000000000
000162   00000000000000000000                00000000000000000000               00000000000000000000
000165   00000000000000000000                00000000000000000000               00000000000000000000
000170   00000000000000000000                00000000000000000000               00000000000000000000
000173   00000000000000000000                00000000000000000000               00000000000000000000
000176   00000000000000000000                00000000000000000000               00000251003640000000     B( 35
000201   00035400000001000001   C=   A   A   00000000004236000006     73    F   00070000004237000616   G    74 FN
000204   00000000004234001412        71 LJ   00000000003075777755       X2||    00000200000001000006    B    A   F
000207   00000000003030000000        XX      77777777777777777762   ||||||||||] 00000000000000000000
000212   17170631463146403615   OOFY-Y-53M   20000000000000000012   P        J  17174000000000422701   OO5    7WA
000215   00000000000261606300        B(=|    05160420000000000000   ENDP        20000000000000000005   P         E
000220   00000000000003540000         C=     00000000000000000000               00000000000000000000
000223   00000503400000040000     EC5  D     00000100000000175000     A    O/   00000000000000000000
000226   00000000000000000000                00000000000000000000               00000000000000000000
000231   57414437551320225700   .694 KPR.    00000000000000000000               14315524111505570000   LY TIME.
000234   33020600000000000000   08F          00000000000000000000               00000000000000000000
000237   00000000000000000000                00000000000000000000               00000000000000000000
000242   00000000000000000000                00000000000000000000               00000000000000000000
000245   00000000000000000000                00000000000000000000               00000000000000000000
000250   55555555364157424240       36.775   55555555553357354242       0.277   55555555553657414437        3.694
000253   00000000000000000000                00000000000000000000               00000000000000000000
```

Example of an incorrect call to DMPABS
LGO (108B, 200B)
ABSOLUTE DUMP FROM 108B TO 200B ARGUMENT ERROR

Figure 23-2. Job 2 (Cont'd)

```
REGDMP  MAIN ROUTINE                                                   COMPASS 3.73309.     74/03/08. 16.34.41.      PAGE      2

                                   IDENT  REGDMP                                                          REGDMP      1
                                   XREF   A                                                               REGDMP      2

                             ***       REGDMP   ODCUMENTATION                                             NUMONE      1
                             *   REGDMP     REGISTER OUMP ROUTINE                                         REGDMP      5
                             *   COPYRIGHT  CONTOL OATA CORPORATION                                       REGDMP      6
                             *  ORIGINAL FORTRAN PLUS COMPASS ROUTINE BY G. NELSON 1972                   REGDMP      7
                             *        ADAPTEO BY M. OMMERMAN MIODLE 1972                                  REGDMP      8
                             * FXTENSIVELY MODIFIEO ANO REWRITTEN ENTIRELY IN COMPASS FOR KRONOS 2.1 REGDMP      9
                             *        BY M. OMMERMAN NOV 1973                                             REGDMP     10
                             * THE ROUTINE LOADEM SAVEM AOAPTEO FROM SCOPE 3.3 SYSTEM.                    REGDMP     11

                             *  THE CALL IS       RJ   REGDMP                                             REGDMP     13
                             *   NO ARGUMENTS ARE PASSEO.                                                 REGDMP     14
                             *   REGISTER CONTENTS ARE OUMPEO TO THE PRINTER INTACT                       REGDMP     15
                             * NO REGISTER IS OESTROYEO ANO ALL REGISTERS ARE RESTOREO TO THE             REGDMP     16
                             *        STATUS BEFORE THE CALL BEFORE CONTROL IS RETURNED TO THE CALLERREGDMP     17
                             *                                                                            REGDMP     18
                             *        THE ROUTINE SAVEM ANO LOAOEM CAN BE USED INDEPENOENT OF REGDMP REGDMP     19
                             *            TO SAVE ANO RESTORE REGISTERS                                   REGDMP     20
                             *    NOTE THAT SAVEM ANO LOAOEM ARE ENTRY POINTS                             REGDMP     21
                             *              HOWEVER, THE VALUES ARE SAVEO IN REGS DEFINED IN REGDMP       REGDMP     22
                             *        A USER CAN SAVE ALL THE REGISTERS BY  RJ SAVEM                      REGDMP     23
                             *          AT SOME LATER POINT IN TIME THE REGISTERS CAN BE RESTORED         REGDMP     24
                             *      BY      RJ  LOAOEM                                                    REGDMP     25
                             *         NOTE EACH NEW CALL TO SAVEM OR REGDMP WILL DESTROY THE LAST        REGDMP     26
                             *  SAVED REGISTERS.                                                          REGDMP     27
                             *    ALSO NOTE.      AFTER A CALL TO SAVEM THE REGISTERS ARE OESTROYED       REGDMP     28
                             *    AND IF THE USER WISHES TO CONTINUE WITH THE SAVE REGISTER VALUES        REGDMP     29
                             *  IT WILL OE NECESSARY TO MAKE A CALL TO LOADEM                             REGDMP     30

                                  ENTRY  REGDMP                                                           REGDMP     32
     0                    30    REGS      BSS    24                                                       REGDMP     33
    30                     1    REGDMP    BSSZ   1                                                        REGDMP     34
    31  0100000317 +                      RJ     =XSAVEM                                                  REGDMP     35
    32  0100000035 +                      RJ     REGDMP1                                                  NUMDNE      2
    33  0100000421 +                      RJ     =XLOADEM                                                 REGDMP     37
    34  0400000030 +                      EQ     REGDMP                                                   REGDMP     38
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP

```
   35                   1      RFGDMP1  BSS    1              REGDMP CALLED FROM                    NUMDNE       3
   36  5110000030 +                     SA1    =XREGDMP                                             REGDMP      41
                 20136                  LX1    30                                                   REGDMP      42
   37  7211777776                       SX1    X1-1           GET CALLERS ADDRESS                   REGDMP      43
                 6110000001             SB1    1               CONVERSION ROUTINES DEMAND  B1 = 1   REGDMP      44
   40  56000                            SA0    B0             SET LOWER 24 BITS TO MACHINE ZERO     REGDMP      45
            0100000272 +                RJ     REGDMP5        B1 WILL BE SET TO 1                   NUMDNE       4
   41  5160000073 +                     SA6    IADD                                                 REGDMP      47
                 6170000010             SB7    8                                                    REGDMP      48
   42  6150000070                       SB5    56                                                   REGDMP      49
                 6140000007             SB4    7                                                    REGDMP      50
   43  5100555555                       SA0    5555558        PROPIGATE SPACES                      REGDMP      51
   44  5114000000 +            REGDMP2  SA1    REGS+B4        GET B REGS                            NUMDNE       5
                 0100000272 +           RJ     REGDMP5                                              NUMDNE       6
   45  5165000075 +                     SA6    BREG+B5                                              REGDMP      54
                 67441                  SB4    B4-B1                                                REGDMP      55
                      67557             SB5    B5-B7          GET TO NEXT LOWER B                   REGDMP      56
   46  0640000044 +                     GE     B4,REGDMP2                                           NUMDNE       7
                 6170000010             SB7    8                                                    REGDMP      58
   47  6150000070                       SB5    56                                                   REGDMP      59
                 6140000007             SB4    7                                                    REGDMP      60
   50  5114000010 +            REGDMP3  SA1    REGS+8+B4                                            NUMDNE       8
                 0100000272 +           RJ     REGDMP5                                              NUMDNE       9
   51  5165000077 +                     SA6    AREG+B5                                              REGDMP      63
                 67441                  SB4    B4-B1                                                REGDMP      64
                      67557             SB5    B5-B7                                                REGDMP      65
   52  0640000050 +                     GE     B4,REGDMP3                                           NUMDNE      10
                 6170000010             SB7    8                                                    REGDMP      67
   53  6150000070                       SB5    56                                                   REGDMP      68
                 6140000007             SB4    7                                                    REGDMP      69
   54  5114000020 +            REGDMP4  SA1    REGS+16+B4                                           NUMDNE      11
                 0100000300 +           RJ     WDD                                                  NUMDNE      12
   55  5165000101 +                     SA6    XREG+B5        TOP OF X                              REGDMP      72
                 5175000102 +           SA7    XREG+1+B5      LOWER PART OF X                       REGDMP      73
   56  67441                            SB4    B4-B1                                                REGDMP      74
            67557                       SB5    B5-B7                                                REGDMP      75
                 0640000054 +           GE     B4,REGDMP4                                           NUMDNE      13
   57  7160000174 +                     SX6    OUT+BUFSIZ                                           REGDMP      77
                 5160000066 +           SA6    OUTPUT+2       READY TO  WRITE                       REGDMP      78
   60  7160000071 +                     SX6    OUT                                                  REGDMP      79
                 5066000001             SA6    A6+1           RESTORE OUT                           REGDMP      80
   61  7120000064 +                     WRITER OUTPUT,R       PUT OUT REG DUMP                      REGDMP      81
   63  0400000035 +                     EQ     REGDMP1                                              NUMDNE      14
   64  17252420252400000001    OUTPUT   VFD    36/6LOUTPUT,24/1                                     REGDMP      84
   65  00000000000000000071 +           VFD    60/OUT                                               REGDMP      85
   66  00000000000000000271 +           VFD    60/OUT+2008                                          REGDMP      86
   67  00000000000000000071 +           VFD    60/OUT                                               REGDMP      87
   70  00000000000000000272 +           VFD    60/OUT+2018                                          REGDMP      88
   71  33220507041520550301    OUT      DATA   20HOREGDMP CALLED FROM                               NUMDNE      15
   73                   1      IADD     BSSZ   1                                                    REGDMP      90
                       75 +    BREG     EQU    *+1                                                  REGDMP      91
                       77 +    AREG     EQU    *+3                                                  REGDMP      92
                      101 +    XREG     EQU    *+5                                                  REGDMP      93
                                        ECHO   ,BM=(B0,B1,B2,B3,B4,B5,B6,B7),AM=(A0,A1,A2,A3,A4,A5,A6,REGDMP      94
                          ,A7),XM=(X0,X1,X2,X3,X4,X5,X6,X7)                                         REGDMP      95
                                        DATA   10H   BM =                                           REGDMP      96
                                        DATA   0                                                    REGDMP      97
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
                                      DATA   10H   AM =                                        REGDMP     98
                                      DATA   0                                                 REGDMP     99
                                      DATA   10H   XM =                                        REGDMP    100
                                      BSSZ   3                                                 REGDMP    101
                                      ENDD                                                     REGDMP    102
                      103    BUFSIZ   EQU    *-OUT                                             REGDMP    103
  174                  76             BSSZ   201B-BUFSIZ FILL OUT BUFFER                       REGDMP    104
  272                   1    REGDMP5  BSS    1           FORMERLY  ADDR                        NUMONE     16
  273  0100000300 +                   RJ     WOD                                               NUMONE     17
  274  43030                          MX0    24                                                NUMONE     18
             15770                    BX7     -X0*X7                                           NUMONE     19
                   20730              LX7    24                                                NUMONE     20
                         74200        SX2    A0                                                NUMONE     21
  275  43052                          MX0    60-18                                             NUMONE     22
             15220                    BX2    -X0*X2                                            NUMONE     23
                   7100000077         SX0    77B                                               NUMONE     24
  276  11020                          BX0    X2*X0       GET ONE EXTRA CHAR                    NUMONE     25
             20206                    LX2    6                                                 NUMONE     26
                   12220              BX2    X2+X0                                             NUMONE     27
                         12672        BX6    X7+X2                                             NUMONE     28
  277  0400000272 +                   EQ     REGDMP5                                           NUMONE     29
  277                        OPL      XTEXT  COMCWOD                                            NUMON  23405
```

Figure 23-3. Job 1 — Create Binary Deck of REGDMP (Cont'd)

```
REGDMP  MAIN ROUTINE                                       COMPASS 3.73309.     74/03/08. 16.34.41.      PAGE      5
LOADEM SAVEM  SAVE AND LOAD ALL REGISTERS
                                 *   COPYRIGHT  CONTOL DATA CORPORATION                        REGDMP      166
                                 * THE ROUTINE LOADEM SAVEM ADAPTED FROM SCOPE 3.3 SYSTEM.    REGDMP      167

                                            ENTRY  SAVEM                                      REGDMP      169
                                            ENTRY  LOADEM                                     REGDMP      170
  316  0100000363 +              RJ         VFD    12/100B,18/BIT0-1,30/1                     REGDMP      171
             0000000001
                                 *$$$$$$$$$$$$$$$$$       REGISTERS ARE SAVED HERE            REGDMP      172
                          0 +     REGISTR   EQU    REGS                                       REGDMP      173
                          0 +     RB0       EQU    REGS                                       REGDMP      174
                          2 +     RB2       EQU    RB0+2                                      REGDMP      175
                          5 +     RB5       EQU    RB0+5                                      REGDMP      176
                          6 +     RB6       EQU    RB5+1                                      REGDMP      177
                          7 +     RB7       EQU    RB6+1                                      REGDMP      178
                         10 +     RA0       EQU    RB7+1                                      REGDMP      179
                         11 +    RA1        EQU    RA0+1                                      REGDMP      180
                         12 +    RA2        EQU    RA0+2                                      REGDMP      181
                         14 +    RA4        EQU    RA0+4                                      REGDMP      182
                         15 +    RA5        EQU    RA0+5                                      REGDMP      183
                         16 +    RA6        EQU    RA0+6                                      REGDMP      184
                         17 +    RA7        EQU    RA0+7                                      REGDMP      185
                         20 +     RX0       EQU    RA7+1                                      REGDMP      186
                         25 +    RX5        EQU    RX0+5                                      REGDMP      187
                         26 +    RX6        EQU    RX0+6                                      REGDMP      188
                         27 +    RX7        EQU    RX0+7                                      REGDMP      189
                                 ***        THESE TWO ROUTINE ARE TAKEN FROM DEBUG AND ARE USED  REGDMP   190
                                 *          TO SAVE AND RESTORE ALL REGISTERS                 REGDMP      191
                                 *                                                            REGDMP      192
  317                     1       SAVEM     BSS    1                                          REGDMP      193
  320  0770000322 +                         NG     B7,BIT17     SAVE B7 WITHOUT TOUCHING ANYTHING ELSE  REGDMP  194
  321  0100000321 +              BITTEN     RJ     *                                          REGDMP      195
             0000000001          -          VFD    30/1                                       REGDMP      196
  322  66777                     BIT17      SB7    B7+B7                                      REGDMP      197
          0770000324 +                      NG     B7,BIT16                                   REGDMP      198
  323  0100000323 +                         RJ     *                                          REGDMP      199
             0000000001          -          VFD    30/1                                       REGDMP      200
  324  66777                     BIT16      SB7    B7+B7                                      REGDMP      201
          0770000326 +                      NG     B7,BIT15                                   REGDMP      202
  325  0100000325 +                         RJ     *                                          REGDMP      203
             0000000001          -          VFD    30/1                                       REGDMP      204
  326  66777                     BIT15      SB7    B7+B7                                      REGDMP      205
          0770000330 +                      NG     B7,BIT14                                   REGDMP      206
  327  0100000327 +                         RJ     *                                          REGDMP      207
             0000000001          -          VFD    30/1                                       REGDMP      208
  330  66777                     BIT14      SB7    B7+B7                                      REGDMP      209
          0770000332 +                      NG     B7,BIT13                                   REGDMP      210
  331  0100000331 +                         RJ     *                                          REGDMP      211
             0000000001          -          VFD    30/1                                       REGDMP      212
  332  66777                     BIT13      SB7    B7+B7                                      REGDMP      213
          0770000334 +                      NG     B7,BIT12                                   REGDMP      214
  333  0100000333 +                         RJ     *                                          REGDMP      215
             0000000001          -          VFD    30/1                                       REGDMP      216
  334  66777                     BIT12      SB7    B7+B7                                      REGDMP      217
          0770000336 +                      NG     B7,BIT11                                   REGDMP      218
  335  0100000335 +                         RJ     *                                          REGDMP      219
             0000000001          -          VFD    30/1                                       REGDMP      220
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
 336  66777                    BIT11     SD7     B7+B7                                        REGDMP
         0770000340 +                    NG      B7,BIT10                                     REGDMP
 337  0100000337 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 340  66777                    BIT10     SB7     B7+B7                                        REGDMP
         0770000342 +                    NG      B7,BIT9                                      REGDMP
 341  0100000341 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 342  66777                    BIT9      SB7     B7+B7                                        REGDMP
         0770000344 +                    NG      B7,BIT8                                      REGDMP
 343  0100000343 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 344  66777                    BIT8      SB7     B7+D7                                        REGDMP
         0770000346 +                    NG      B7,BIT7                                      REGDMP
 345  0100000345 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 346  66777                    BIT7      SB7     D7+B7                                        REGDMP
         0770000350 +                    NG      B7,BIT6                                      REGDMP
 347  0100000347 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 350  66777                    BIT6      SB7     B7+B7                                        REGDMP
         0770000352 +                    NG      B7,BIT5                                      REGDMP
 351  0100000351 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 352  66777                    BIT5      SB7     B7+B7                                        REGDMP
         0770000354 +                    NG      D7,BIT4                                      REGDMP
 353  0100000353 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 354  66777                    BIT4      SB7     B7+B7                                        REGDMP
         0770000356 +                    NG      B7,BIT3                                      REGDMP
 355  0100000355 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 356  66777                    BIT3      SB7     D7+B7                                        REGDMP
         0770000360 +                    NG      B7,BIT2                                      REGDMP
 357  0100000357 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 360  66777                    BIT2      SD7     D7+B7                                        REGDMP
         0770000362 +                    NG      D7,BIT1                                      REGDMP
 361  0100000361 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 362  66777                    BIT1      SD7     B7+B7                                        REGDMP
         0770000364 +                    NG      B7,BIT0                                      REGDMP
 363  0100000363 +                       RJ      *                                            REGDMP
              0000000001    -            VFD     30/1                                         REGDMP
 364  65770                    BIT0      SB7     A7-B0                                        REGDMP
         5170000027 +                    SA7     RX7                                          REGDMP
                   75760                 SX7     A6-B0                                        REGDMP
 365  5170000016 +                       SA7     RA6                                          REGDMP
              5160000026 +               SA6     RX6                                          REGDMP
 366  77770                              SX7     B7-B0                                        REGDMP
         5170000017 +                    SA7     RA7                                          REGDMP
                   10755                 BX7     X5                                           REGDMP
 367  6170000002                         SB7     2                                            REGDMP
              5170000025 +               SA7     RX5                                          REGDMP
 370  10644                              BX6     X4                                           REGDMP
         22703                           LX7     X3                                           REGDMP
              55667                      SA6     A6-B7                                        REGDMP
```

Figure 23-3. Job 1 — Create Binary Deck of REGDMP (Cont'd)

```
                        55777               SA7     A7-B7                                                      REGDMP     278
    371  10622                              BX6     X2                                                         REGDMP     279
                55667                       SA6     A6-B7                                                      REGDMP     280
                     22701                  LX7     X1                                                         REGDMP     281
                          55777             SA7     A7-B7                                                      REGDMP     282
    372  10600                              BX6     X0                                                         REGDMP     283
                55667                       SA6     A6-B7                                                      REGDMP     284
                     75750                  SX7     A5-B0                                                      REGDMP     285
    373  5170000015 +                       SA7     RA5                                                        REGDMP     286
                     75640                  SX6     A4-B0                                                      REGDMP     287
                          75730             SX7     A3-B0                                                      REGDMP     288
    374  5160000014 +                       SA6     RA4                                                        REGDMP     289
                     55777                  SA7     A7-B7                                                      REGDMP     290
                          43373             MX3     59                                                         REGDMP     291
    375  6170000042                         SB7     BIT0-BIT17                                                 REGDMP     292
                     43500                  MX5     0                                                          REGDMP     293
    376  5140000316 +                       SA4     RJ                                                         REGDMP     294
                     14633                  BX6     -X3                                                        REGDMP     295
                          10744             BX7     X4                                                         REGDMP     296
    377  20637                              LX6     31                                                         REGDMP     297
    400  5147000321 +              SAVEM1   SA4     B7+BITTEN                                                  NUMONE      31
                     21501                  AX5     1                                                          REGDMP     299
                          15443             BX4     -X3*X4                                                     REGDMP     300
    401  54740                              SA7     A4                                                         REGDMP     301
                20421                       LX4     17                                                         REGDMP     302
                     6177777775             SB7     B7-2                                                       REGDMP     303
    402  36554                              IX5     X5+X4                                                      REGDMP     304
                37776                       IX7     X7-X6                                                      REGDMP     305
                     0670000400 +           PL      B7,SAVEM1                                                  NUMONE      32
    403  75610                              SX6     A1-B0                                                      REGDMP     307
                75720                       SX7     A2-B0                                                      REGDMP     308
                     5170000012 +           SA7     RA2                                                        REGDMP     309
    404  7120000004                         SX2     4                                                          REGDMP     310
                     6170000001             SB7     1                                                          REGDMP     311
    405  43052                              MX0     42                                                         REGDMP     312
                55677                       SA6     A7-B7                                                      REGDMP     313
                     75700                  SX7     A0-B0                                                      REGDMP     314
    406  5130000415 +                       SA3     A0B7M                                                      REGDMP     315
                     43173                  MX1     59                                                         REGDMP     316
    407  22605                     SAVEM2   LX6     X5                                                         NUMONE      33
                55767                       SA7     A6-B7                                                      REGDMP     318
                     15450                  BX4     -X0*X5                                                     REGDMP     319
                          36221             IX2     X2+X1                                                      REGDMP     320
    410  55677                              SA6     A7-B7                                                      REGDMP     321
                77550                       SX5     B5-B0                                                      REGDMP     322
                     15170                  BX1     -X0*X7                                                     REGDMP     323
                          36434             IX4     X3+X4                                                      REGDMP     324
    411  54337                              SA3     A3+B7                                                      REGDMP     325
                67530                       SB5     B3-B0                                                      REGDMP     326
                     20436                  LX4     30                                                         REGDMP     327
                          36741             IX7     X4+X1                                                      REGDMP     328
    412  43173                              MX1     59                                                         REGDMP     329
                67310                       SB3     B1-B0                                                      REGDMP     330
                     5272000454 +           SA7     RESTB+X2                                                   REGDMP     331
    413  77460                              SX4     B6-B0                                                      REGDMP     332
                67640                       SB6     B4-B0                                                      REGDMP     333
                     67420                  SB4     B2-B0                                                      REGDMP     334
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
REGDMP  MAIN ROUTINE                                        COMPASS 3.73309.     74/03/08. 16.34.41.      PAGE      8
LOADEM SAVEM  SAVE AND LOAD ALL REGISTERS

                     10744            BX7     X4                                                  REGDMP     335
   414  0312000407 +                  NZ      X2,SAVEM2                                           NUMONE      34
                0400000317 +          EQ      SAVEM                                               REGDMP     337
   415  5000000000          A0B7M     SA0     A0+0                                                REGDMP     338
                6070000000            SB7     A0+0                                                REGDMP     339
   416  6060000000                    SB6     A0+0                                                REGDMP     340
                6050000000            SB5     A0+0                                                REGDMP     341
   417  6040000000                    SB4     A0+0                                                REGDMP     342
                6030000000            SB3     A0+0                                                REGDMP     343
   420  6020000000                    SB2     A0+0                                                REGDMP     344
                6010000000            SB1     A0+0                                                REGDMP     345
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
                                  *     R E S T O R E   R E G I S T E R S                     REGDMP    347
  421                    1      LOADEM  BSS    1                                                REGDMP    348
  422  6110000001                       SB1    1                                                REGDMP    349
                 6140000003             SB4    3                                                REGDMP    350
  423  66244                            SB2    B4+B4                                            REGDMP    351
            66300                       SB3    B0                                               REGDMP    352
                 43052                  MX0    42                                               REGDMP    353
  424  5113000415 +             RESTA   SA1    B3+A0B7M                                         NUMONE     35
                 5122000001 +           SA2    B2+RB0+1                                         REGDMP    355
  425  66331                            SB3    B3+B1                                            REGDMP    356
            54321                       SA3    A2+B1                                            REGDMP    357
                 15220                  BX2    -X0*X2                                           REGDMP    358
                      15330             BX3    -X0*X3                                           REGDMP    359
  426  36621                            IX6    X2+X1                                            REGDMP    360
            6122777775                  SB2    B2-2                                             REGDMP    361
                      20636             LX6    30                                               REGDMP    362
  427  36663                            IX6    X6+X3                                            REGDMP    363
            5164000454 +                SA6    B4+RESTB                                         REGDMP    364
                      67441             SB4    B4-B1                                            REGDMP    365
  430  0640000424 +                     PL     B4,RESTA                                         NUMONE     36
                 5110000017 +           SA1    RA7                                              REGDMP    367
  431  6170000001                       SB7    1                                                REGDMP    368
                 53210                  SA2    X1                                               REGDMP    369
                      10722             BX7    X2                                               REGDMP    370
  432  55317                            SA3    A1-B7                                            REGDMP    371
            53430                       SA4    X3                                               REGDMP    372
                 22604                  LX6    X4                                               REGDMP    373
  433  5271777777                       SA7    X1+777777B                                       REGDMP    374
                 5263777777             SA6    X3+777777B                                       REGDMP    375
  434  5110000027 +                     SA1    RX7                                              REGDMP    376
                 55217                  SA2    A1-B7                                            REGDMP    377
                      10711             BX7    X1                                               REGDMP    378
  435  22602                            LX6    X2                                               REGDMP    379
            55337                       SA3    A3-B7                                            REGDMP    380
                 55127                  SA1    A2-B7                                            REGDMP    381
  436  5253777777                       SA5    X3+777777B                                       REGDMP    382
                 55337                  SA3    A3-B7                                            REGDMP    383
                      55217             SA2    A1-B7                                            REGDMP    384
  437  10511                            BX5    X1                                               REGDMP    385
            5243777777                  SA4    X3+777777B                                       REGDMP    386
                      55337             SA3    A3-B7                                            REGDMP    387
  440  55127                            SA1    A2-B7                                            REGDMP    388
            10422                       BX4    X2                                               REGDMP    389
                 55237                  SA2    A3-B7                                            REGDMP    390
  441  5233777777                       SA3    X3+777777B                                       REGDMP    391
                 5222777777             SA2    X2+777777B                                       REGDMP    392
  442  10311                            BX3    X1                                               REGDMP    393
            55117                       SA1    A1-B7                                            REGDMP    394
                 10211                  BX2    X1                                               REGDMP    395
                      55117             SA1    A1-B7                                            REGDMP    396
  443  26011                            UX0    B1,X1                                            REGDMP    397
            20013                       LX0    11                                               REGDMP    398
                 26120                  UX1    B2,X0                                            REGDMP    399
                      20113             LX1    11                                               REGDMP    400
  444  26031                            UX0    B3,X1                                            REGDMP    401
            20013                       LX0    11                                               REGDMP    402
                 26140                  UX1    B4,X0                                            REGDMP    403
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
REGDMP  MAIN ROUTINE                                          COMPASS 3.73309.     74/03/08. 16.34.41.      PAGE    10
LOADEM SAVEM   SAVE AND LOAD ALL REGISTERS

                 43074               MX0    60                                                            REGDMP    404
  445  5200777777                    SA0    X0+777777B                                                    REGDMP    405
            20113                    LX1    11                                                            REGDMP    406
                 26051               UX0    B5,X1                                                         REGDMP    407
  446  55117                         SA1    A1-B7                                                         REGDMP    408
       20013                         LX0    11                                                            REGDMP    409
            6270777777               SB7    X0+777777B                                                    REGDMP    410
  447  26060                         UX0    B6,X0                                                         REGDMP    411
       10011                         BX0    X1                                                            REGDMP    412
            5110000011 +             SA1    RA1                                                           REGDMP    413
  450  5211777777                    SA1    X1+777777B                                                    REGDMP    414
            77170                    SX1    B7-B0                                                         REGDMP    415
                 27161               PX1    B6,X1                                                         REGDMP    416
  451  21113                         AX1    11                                                            REGDMP    417
       27151                         PX1    B5,X1                                                         REGDMP    418
            21113                    AX1    11                                                            REGDMP    419
                 27141               PX1    B4,X1                                                         REGDMP    420
  452  21113                         AX1    11                                                            REGDMP    421
       27131                         PX1    B3,X1                                                         REGDMP    422
            21113                    AX1    11                                                            REGDMP    423
                 27121               PX1    B2,X1                                                         REGDMP    424
  453  21113                         AX1    11                                                            REGDMP    425
       27111                         PX1    B1,X1                                                         REGDMP    426
  454  00000000000000000000   RESTB   DATA   0,0,0,0                                                       REGDMP    427
  460  0400000421 +                  EQ     LOADEM                                                        REGDMP    428
  461                                END                                                                  REGDMP    429

            42200B CM   STORAGE USED          479 STATEMENTS        62 SYMBOLS
                        MODEL 73 ASSEMBLY   3.667 SECONDS          157 REFERENCES
```

An example of the control cards to create the binaries.

```
MORRI, CM60000, T7777.
ACCOUNT, ML0.
COMPASS.
SAVE(LGO=REGDMP)
```

An example of the output from a call to REGDMP is:

```
REGDMP CALLED FROM 000574
  B0 =    000000       A0 =    000700       X0 =    00000000000000000000
  B1 =    000001       A1 =    000561       X1 =    55555555340000000001
  B2 =    000002       A2 =    000562       X2 =    55555502350000000002
  B3 =    005407       A3 =    000563       X3 =    10030405360000000035
  B4 =    000000       A4 =    000000       X4 =    00000000000000000000
  B5 =    000000       A5 =    000000       X5 =    00000000000000000000
  B6 =    057752       A6 =    000001       X6 =    20061420000573000700
  B7 =    000000       A7 =    000000       X7 =    00000000000000000000
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP (Cont'd)

```
                                            IDENT   TEST
                                            ENTRY   TEST
  0  55555555340000000001       ONE         VFD      30/5H    1,30/1
  1  55555502350000000002       TWO         VFD      30/5H   B2,30/2
  2  10030405360000000035       THREE       VFD      30/5HHCOE3,30/35B
  3  01234567010000000000       TOP         VFD     30/01234567018,30/0
  4  00000000000123456701       BOTTOM      VFD     30/0,30/01234567018
  5  01234567012345670123       ALL         VFD     60/01234567012345670123B
  6  00000000000000000010       EIGHT       VFD     60/8
  7  17235314631463146315       ONEPT       DATA    10.8
 10  17204000000000000000       FLO         DATA    1.0
 11  16566676337663536756       TENTEN      DATA    1.0E-10
 12                             TEST        BSS     0
 12  5110000000 +                           SA1     ONE
               5120000001 +                 SA2     TWO
 13  5130000002 +                           SA3     THREE
               0100000000 X                 RJ      =XREGDMP
 14  6110000001                             SB1     1
               6120000002                   SB2     2
 15  6130000003                             SB3     3
               6140000004                   SB4     4
 16  6150000005                             SB5     5
               6160000006                   SB6     6
 17  6170000007                             SB7     7
               0100000000 X                 RJ      =XREGDMP
 20  0100000000 X                           RJ      =XREGDMP
 21  5140000003 +                           SA4     TOP
               5150000004 +                 SA5     BOTTOM
 22  73610                                  SX6     X1
          73720                             SX7     X2
               5110000012 +                 SA1     TEST
 23  0100000000 X                           RJ      =XREGDMP
 24  5160000001 +                           SA6     TWO
               5170000000 +                 SA7     ONE
 25  0100000000 X                           RJ      =XREGDMP
 26  5100000005                             SA0     5
               5110000000 +                 SA1     ONE
 27  5120000001 +                           SA2     TWO
               0100000000 X                 RJ      =XREGDMP
 30  5110000005 +                           SA1     ALL
               5130000007 +                 SA3     ONEPT
 31  5120000006 +                           SA2     EIGHT
               5140000010 +                 SA4     FLO
 32  5150000011 +                           SA5     TENTEN
               63110                        SB1     X1
                    63220                  SB2      X2
 33  63330                                  SB3      X3
          63440                            SB4      X4
               63550                         SB5     X5
                    64610                   SB6     A1
 34  64720                                  SB7     A2
          56000                             SA0     B0
               0100000000 X                 RJ      =XREGDMP
 35  7160247021                             ENDRUN
 37                                         END     TEST

                   41600B CM  STORAGE USED              58 STATEMENTS       13 SYMBOLS
```

Figure 23-3. Job 1 – Create Binary Deck of REGDMP
(Example of Job Using REGDUMP)

```
         IDENT     BOB, 101B, BOB
         ABS
         SST

***      BOB - USE MONITOR FUNCTION RSB TO READ OUR EST AT 6600B

         ENTRY     BOB
         ENTRY     DMP=
         ENTRY     RFL=
         ENTRY     SSJ=

         ORG       101B
DMP=     EQU       0
SSJ=     VFD       12/0, 24/-0, 12/MPRS-10, 12/MXPS+1
         BSSZ      4

BOB      SA1       CALL
         BX6       X1
         SA6       1
  +      SA1       1
         NZ        X1, *
         SX7       3REND
         LX7       42
         SA7       1
         PS

CALL     VFD       24/0LRSBP, 18/0, 18/SS
SS       VFD       12/0, 12/100B, 18/6600B, 18/BUF
BUF      CON       -0
         BSSZ      100B
RFL=     EQU       *
         END

34 STATEMENTS           266 SYMBOLS
0.231 SECONDS            17 REFERENCES
```

Figure 23-4. CP Routine Using Special Entry Points and Monitor Function RSB.

```
OPR - REAO SECTOR                                    COMPASS 3.73213     73/12/19. 13.51.09.     PAGE    2

                                  IOENT  DPR,PPFW
                                  PERIPH
                  D*M             BASE   MIXED
                                  SST

                       ***        OPR - READ A SPECIFIC SECTOR
                       *
                       *          R. J. ENGBERG         73/12/01


                       ***        CALL - MADE UNOER OIS (USUALLY) WITH FOLLOWING FORMAT -
                       *
                       *                 OPR(XXXX,YYZZZ)
                       *          WHERE,
                       *                 XXXX = TRACK NUMBER
                       *                 YY   = EST ORDINAL OF MASS STORAGE DEVICE
                       *                 ZZZ  = SECTOR NUMBER TO BE READ
                       *
                       *          EXIT INFORMATION - THE SECTOR IS WRITTEN AT RA+100B.  RA+77B
                       *                 CONTAINS THE HEADER BYTES IN FORMAT -
                       *                 12/EQ,12/TK,12/SE,12/B1,12/B2
                       *                 WHERE -
                       *                 EQ = EST ORDINAL
                       *                 TK = TRACK NUMBER
                       *                 SE = SECTOR NUMBER
                       *                 B1 = HEADER BYTE 1
                       *                 B2 = HEAOER BYTE 2

                       **         THE PP INPUT REGISTER FORMAT IS -
                       *
                       *T IR      VFO    18/ DPR,6/,18/ TK ,9/EQ,9/ SE
                       *
                       *          WHERE -
                       *                 TK = TRACK NUMBER
                       *                 EQ = EST ORDINAL
                       *                 SE = SECTOR NUMBER


                       *          MEMORY CELLS
                  20    CH        EQU    20          CHANNEL NUMBER
                6775    EQ        EQU    BFMS-3      EST ORDINAL
                6776    TK        EQU    BFMS-2      TRACK
                6777    SE        EQU    BFMS-1      SECTOR
```

Figure 23-5.   DPR-PPFW Read Specific Sector

```
DPR - READ SECTOR                                    COMPASS 3.73213    73/12/19. 13.51.09.     PAGE    3
                                                     DPR

1100                              ORG    PPFW

1100       1400         DPR       LDN    0
1101       6010                   CRD    CM
1102       3074                   LDD    CP
1103       1635                   ADN    MS2W
1104       6210                   CWD    CM             CLEAR LAST MESSAGE
1105       3056                   LDD    FL             CHECK FIELD LENGTH
1106       1702                   SBN    2
1107       0605                   PJN    DPR1           200B CM WORDS OR MORE
1110       2000 1302              LDC    =C* FL TOO SHORT - DPR.*
1112       0100 1273              LJM    ERR            ABORT

1114       3054         DPR1      LDD    IR+4           PICK UP SECTOR NUMBER
1115       2200 0777              LPC    777            BITS 0-8
1117       5400 6777              STM    SE
1121       3054                   LDD    IR+4           PICK UP EST ORDINAL
1122       1066                   SHN    -11            BITS 9-11
1123       5400 6775              STM    EQ
1125       3053                   LDD    IR+3
1126       1014                   SHN    14
1127       5400 6776              STM    TK             SAVE BITS 0-5 OF TK NUMBER
1131       1066                   SHN    -11
1132       1270                   LPN    70             BITS 12-14 = LEFT HAND 3 BITS OF EQ
1133       5500 6775              RAM    EQ
1135       3052                   LDD    IR+2
1136       1277                   LPN    77
1137       1006                   SHN    6
1140       5500 6776              RAM    TK             SAVE TK

                        *         CHECK FOR MASS STORAGE DEVICE

1142       5000 0551              LDM    ESTS           FWA OF EST
1144       5100 6775              ADM    EQ
1146       6010                   CRD    CM
1147       3010                   LDD    CM
1150       1006                   SHN    6
1151       0705                   MJN    DPR2
1152       2000 1315              LDC    =C* NOT MASS STORAGE DEVICE - DPR.*
1154       0100 1273              LJM    ERR

1156       3014         DPR2      LDD    CM+4           MST ADDRESS/10B
1157       1003                   SHN    3
                        *         ADN    TRTL
1160       6000                   CRD    T0
1161       1601                   ADN    1
1162       6005                   CRD    T5
1163       3002                   LDD    T2             LENGTH OF TRT IN CM WORDS
1164       1002                   SHN    2              *4 = NUMBER OF TRACKS
1165       5200 6776              SBM    TK
1167       0605                   PJN    DPR3           REQUESTED TRACK VALID
1170       2000 1336              LDC    =C* ILLEGAL TRACK - DPR.*
1172       0100 1273              LJM    ERR

1174       3010         DPR3      LDD    T7+1
1175       2200 7700              LPC    7700
1177       1006                   SHN    6
```

Figure 23-5.   DPR-PPFW Read Specific Sector

```
1200    3107                 ADD   T7
1201    1006                 SHN   6           NUMBER OF SECTORS/TRACK
1202    5200 6777            SBM   SE
1204    1701                 SBN   1
1205    0605                 PJN   DPR4        REQUESTED SECTOR VALID
1206    2000 1352            LDC   =C* ILLEGAL SECTOR - DPR.*
1210    0100 1273            LJM   ERR

                     *       READ SECTOR FROM DISK

1212    5000 6776    DPR4    LDM   TK
1214    3406                 STD   T6          SET TRACK
1215    5000 6777            LDM   SE
1217    5400 0007            STM   T7          SET SECTOR
1221    5000 6775            LDM   EQ
1223    3405                 STD   T5          SET EST ORDINAL
1224    0200 0547            RJM   SMS         SET DRIVER
1226    3011                 LDD   CH+1        CHANNEL NUMBER
1227    3420                 STD   CH
1230    0200 0437            RJM   RCH         RESERVE CHANNEL
1232    0200 0606            RJM   POS         POSITION DISK
1234    2000 7000            LDC   BFMS        FWA OF BUFFER
1236    0200 0616            RJM   RDS         READ SECTOR
1240    3020                 LDD   CH
1241    0200 0446            RJM   DCH         RELEASE CHANNEL

                     *       STORE SECTOR IN RA+100B AND HEADER IN RA+77B

1243    2000 0101            LDC   101
1245    3401                 STD   T1
1246    3055                 LDD   RA          RA/100B
1247    1006                 SHN   6           RA
1250    1677                 ADN   77          RA+77
1251    6301 6775            CWM   BFMS-3,T1

                     *       TERMINATE

1253    3074                 LDD   CP
1254    1635                 ADN   MS2W        USE SECOND LINE
1255    6370 1264            CWM   MSG,ON
1257    1444                 LDN   DPPM
1260    0200 0364            RJM   FTN
1262    0100 0103            LJM   PPR


1264    0420         MSG     DIS   ,*DPR COMPLETE*



1273    0200 0501    ERR     RJM   DFM         SEND DAYFILE MESSAGE
1275    1436                 LDN   ABTM
1276    0200 0364            RJM   FTN
1300    0100 0103            LJM   PPR

1366                         END
```

Figure 23-5. DPR-PPFW Read Specific Sector

```
DPR - READ SECTOR                                                      COMPASS 3.73213      73/12/19. 13.51.09.      PAGE      4
                                                                       OPR

1200          3107                      ADD     T7
1201          1006                      SHN     6              NUMBER OF SECTORS/TRACK
1202          5200 6777                 SBM     SE
1204          1701                      SBN     1
1205          0605                      PJN     OPR4           REQUESTED SECTOR VALID
1206          2000 1352                 LDC     =C* ILLEGAL SECTOR - DPR.*
1210          0100 1273                 LJM     ERR

                               *        READ SECTOR FROM DISK

1212          5000 6776         OPR4    LDM     TK
1214          3406                      STD     T6             SET TRACK
1215          5000 6777                 LDM     SE
1217          5400 0007                 STM     T7             SET SECTOR
1221          5000 6775                 LDM     EQ
1223          3405                      STD     T5             SET EST ORDINAL
1224          0200 0547                 RJM     SMS            SET DRIVER
1226          3011                      LDD     CM+1           CHANNEL NUMBER
1227          3420                      STD     CH
1230          0200 0437                 RJM     RCH            RESERVE CHANNEL
1232          0200 0606                 RJM     POS            POSITION DISK
1234          2000 7000                 LDC     BFMS           FWA OF BUFFER
1236          0200 0616                 RJM     RDS            READ SECTOR
1240          3020                      LDD     CH
1241          0200 0446                 RJM     DCH            RELEASE CHANNEL

                               *        STORE SECTOR IN RA+100B AND HEADER IN RA+77B

1243          2000 0101                 LDC     101
1245          3401                      STD     T1
1246          3055                      LDD     RA             RA/100B
1247          1006                      SHN     6              RA
1250          1677                      ADN     77             RA+77
1251          6301 6775                 CWM     BFMS-3,T1

                               *        TERMINATE

1253          3074                      LDD     CP
1254          1635                      ADN     MS2W           USE SECOND LINE
1255          6370 1264                 CWM     MSG,ON
1257          1444                      LDN     DPPM
1260          0200 0364                 RJM     FTN
1262          0100 0103                 LJM     PPR


1264          0420              MSG     DIS     ,*DPR COMPLETE*



1273          0200 0501         ERR     RJM     DFM            SEND DAYFILE MESSAGE
1275          1436                      LDN     ABTM
1276          0200 0364                 RJM     FTN
1300          0100 0103                 LJM     PPR

1366                                    END
```

Figure 23-5.   DPR-PPFW Read Specific Sector

DEADSTART 24

24.0 INTRODUCTION

The KRONOS operating system consists of PP programs, CP programs, macro definitions, and symbol definitions. The entire system is contained in a magnetic tape file produced by the library maintenance routine MODIFY. Programs in the library file are in source language form. Installation options are provided to permit flexible selection of system features during the assembly and creation of a deadstart file on tape.

The deadstart (DS) tape is a collection of binary files created by the library maintenance routine LIBEDIT. The DS tape is used to start up a CDC CYBER 70 or 6000 Series computer and load the KRONOS operating system.

To load the operating system into a CDC CYBER 70 or 6000 Series computer, the DS tape is mounted on a device (magnetic tape drive), and a small bootstrap loader program is set up on the hardware deadstart panel switches. The deadstart procedures are explained in Part II of the Installation Handbook.

24.1 HARDWARE DEADSTART

When the operator hits the DS button, the following occurs:

1) The DS panel (Table 24-1) is input across channel 0 into PP0 locations 1 through 14. The DS controller will output 1 byte of zeros and then the DS panel to PP0. It then issues a DCN and PPO begins executing at loc (P)+1 = 0+1 = 1. PP0 is ready to perform an IAM at DS.

2) Each PPU except PP0 is connected to its corresponding channel (i.e., PP1 connects to channel 1, PP2 connects to channel 2, etc.).

3) Channels 0, 12, and 13 are not connected (that is why the tape channel is wired as channel 13).

4) The (A) of each PP is set=10000B so that a PP can input its entire field length before automatically disconnecting from the channel.

5) Each PP will hang on an IAM on its channel. This simulates the PP contents bytes 0, 1, 2, and 3 as follows (See figure 24-1):

0	0000	start at location P+1 = 1
1	1500	LCN 0 set (A) = 777777
2	71pp	IAM pp number = channel number
3	0000	at location 0

In actuality, the hardware will set (A) = 10000, and place the PP in TRIP 4 mode of the IAM instruction. Bytes 1, 2, and 3 are not destroyed.

When a PP dump is requested, the following 4 bytes are sent to the PP.

0	0000	start at location 1
1	1500	LCN 0 set (A) = 777777
2	73pp	OAM pp number = channel number
3	0000	output on channel pp all of PP memory.

Thus bytes 0, 1, 2, and 3 are lost in the dump.

Figure 24-1 is a description of the IAM instruction.

6) PP0 will begin executing at location 1.

7) The CPU will do a hardware idle.

When the IAM begins, (P) is stored in location 0. As each 12-bit PP word is transmitted across the channel, (A) is decremented by 1. Whenever (A)=0 or the channel is disconnected by another PP or a controller, the receiving PP will store (0)+1 into (P) and execution begins at the location thus formed. This property is used to autoload routines. A PP will IAM a set of words and then input as the final word an execution address minus one into location zero. The PP will then begin executing at the execution address specified (see SFP in PP resident Section 4). PPs may communicate with a piece of hardware via a channel or with another PP via a channel. If a PP communicates with some hardware, it must set its (A) to the number of words it wishes to input. When this number of words has been input, it will execute at (0)+1. If two PPs are communicating when the transmitting PP does a DCN on the channel, the receiving PP will begin executing as if (A) went to zero.

TABLE 24-1. DEADSTART PANEL

Location	Binary	PP COMPASS	Description
1	75 13	DCN 13	disconnect channel 13
2	77 13	FNC 13, e00u	function on channel 13
3	e0 0u	eqe Unit u	equipment e unit u
4	77 13	FNC 13, 0010	function on channel 13
5	00 10	rewind code	rewind
6	77 13	FNC 13, 1400	function on channel 13
7	14 00	select input to eor	select read to eor
10	74 13	ACN 13	activate channel 13
11	71 13	IAM 13, 6606	input from channel 13
12	66 06	load address	to PP loc 6606
13	0XXY	–	see Part II, Section 2 of the Installation Handbook
14	RPSS	–	see Part II, Section 2 of the Installation Handbook

24.2 SOFTWARE DEADSTART

This section describes both the Phase I deadstart and Phase II system activation.

24.2.1 Phase I Start Up

When the hardware is ready, PP0 will begin executing the program on the deadstart panel. Refer to Table 24-1, Figure 24-2, and the system catalog in Section III of the Installation Handbook.

1) PP0 will disconnect channel 13 (clear it), then connect (via function) to channel 13, equipment e, unit u, and rewind the DS tape. It will then read the first record from the DS tape into its memory starting at location 6606. This record is the binary deck PRL "system tape preloader." PRL is 1053B bytes

NOTE

This instruction will hang up the Peripheral Processor if executed when the channel is inactive.

71 IAM md Input (A) words to m from channel d (24 Bits)

f	d	m
25 18	17 12	11 0
(P)		(P+1)

This instruction transfers a block of 12-bit words from input channel d to the processor memory. The content of A gives the block length. The contents of location m specifies the processor address which is to receive the first word. The content of A is reduced by one as each word is read. The input operation is complete when A = 0 or the data channel becomes inactive. If the operation is terminated by the channel becoming inactive, the next location in the processor memory is set to all zeroes. However, the word count is not affected by this empty word. Therefore, the contents of the A register gives the block length minus the number of real data words actually read in.

During this instruction address 0000 temporarily holds P, while m is held in the P register. The content of P advances by one to give the address for the next word as each word is stored.

Figure 24-1. The IAM Instruction

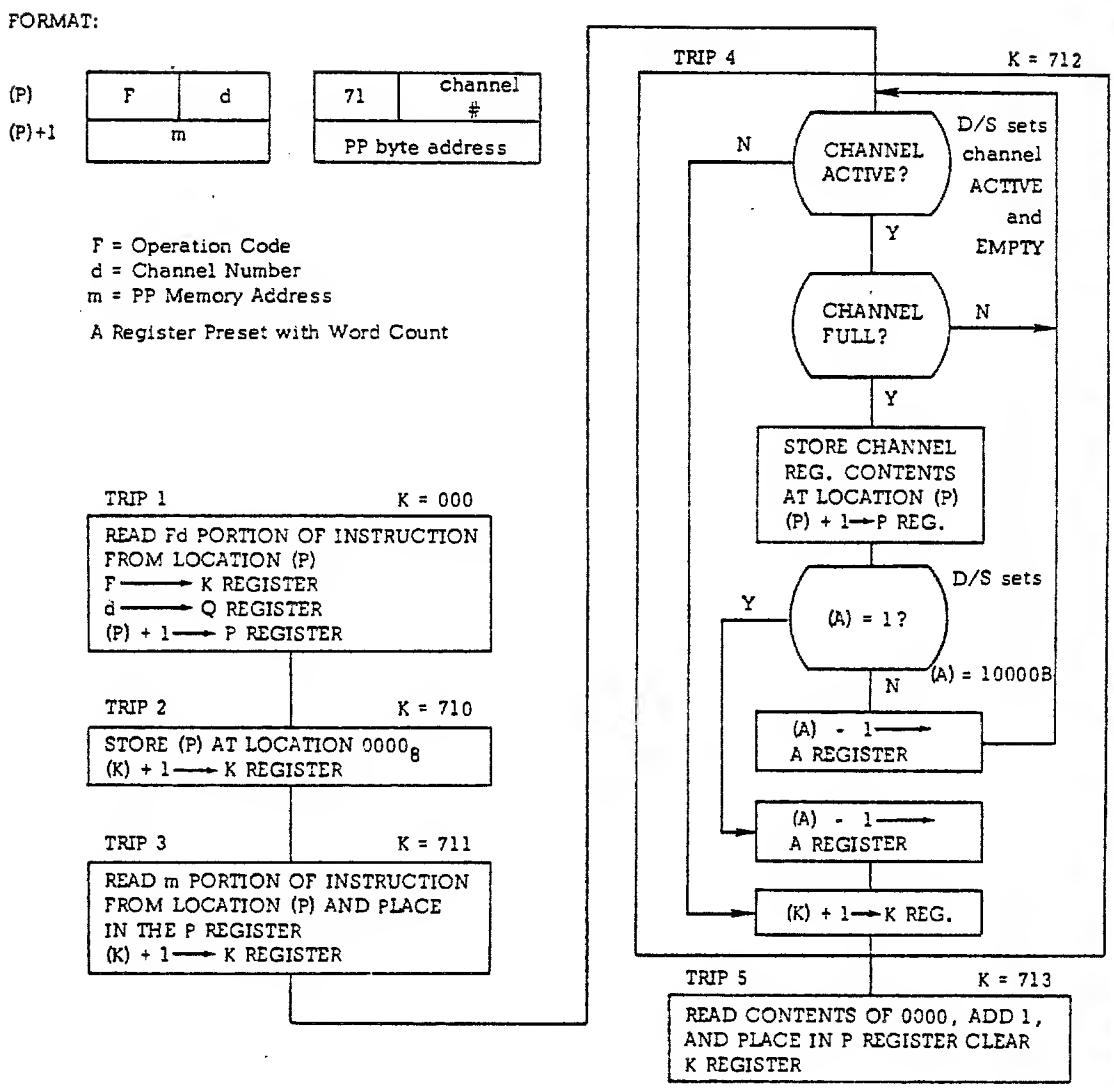
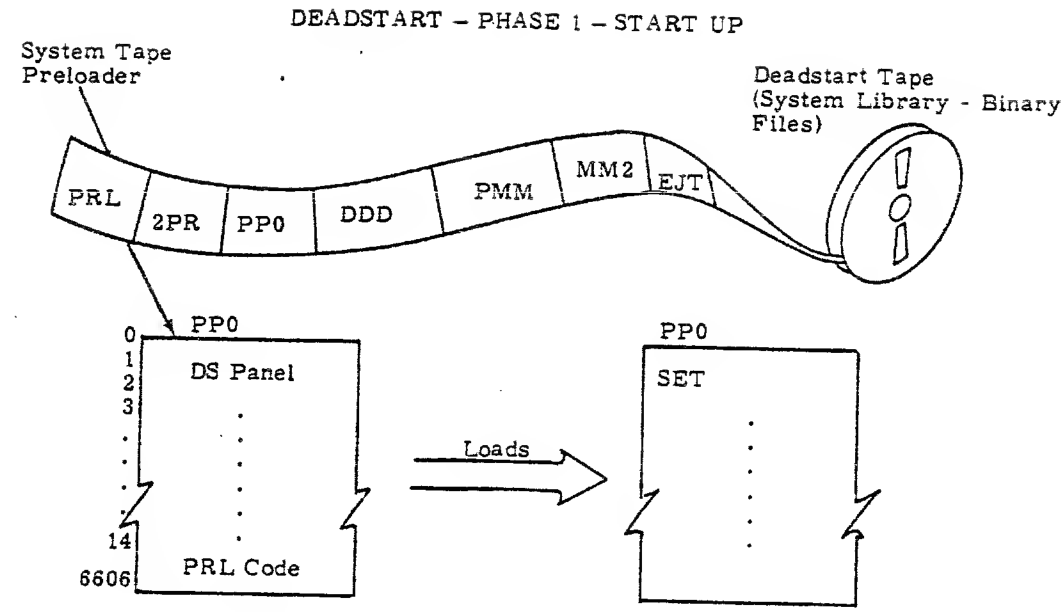


Figure 24-1. The IAM Instruction (Continued)



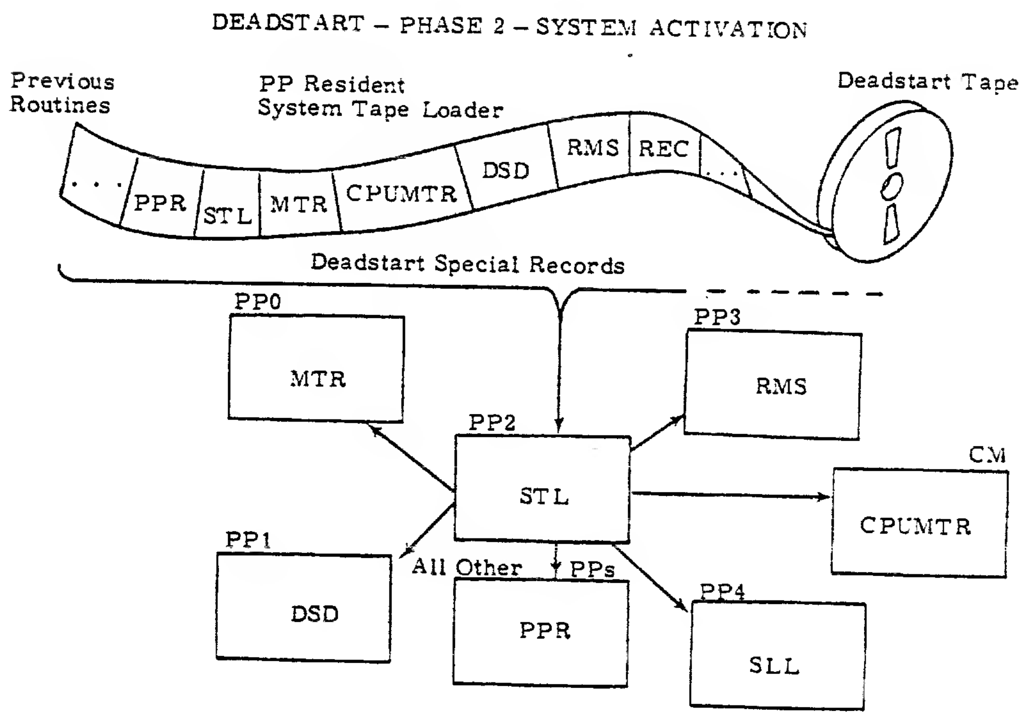
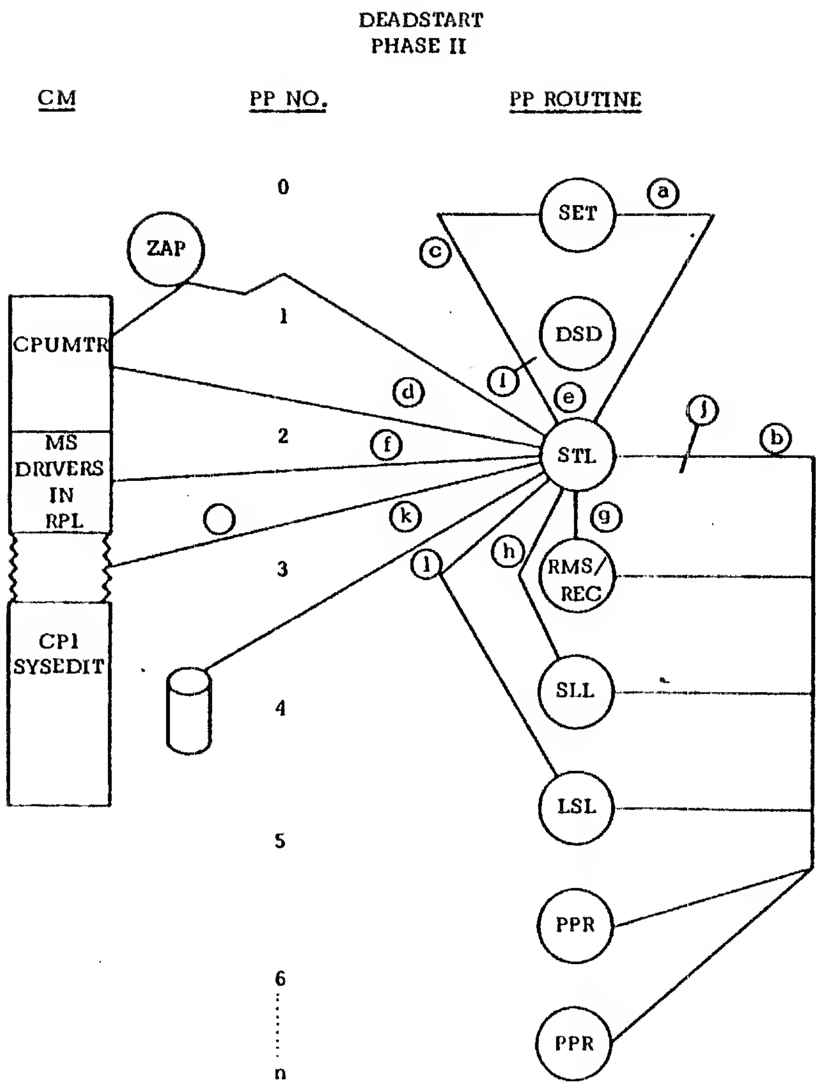


Figure 24-2. DS Tape



DESCRIPTION

a. SET load PPR and STL into PP2.

b. STL loads PPR into all PPs except PP0 and PP1.

c. Load MTR into PP0.

d. Load and initialize CPUMTR.

e. Load DSD into PP1.

f. Load MS drivers into RPL.

g. Load RMS to PP3.

h. Load SLL to PP4 if system tape load required.

i. Disconnect CRO and activate MTR.

j. Activate all other PPs.

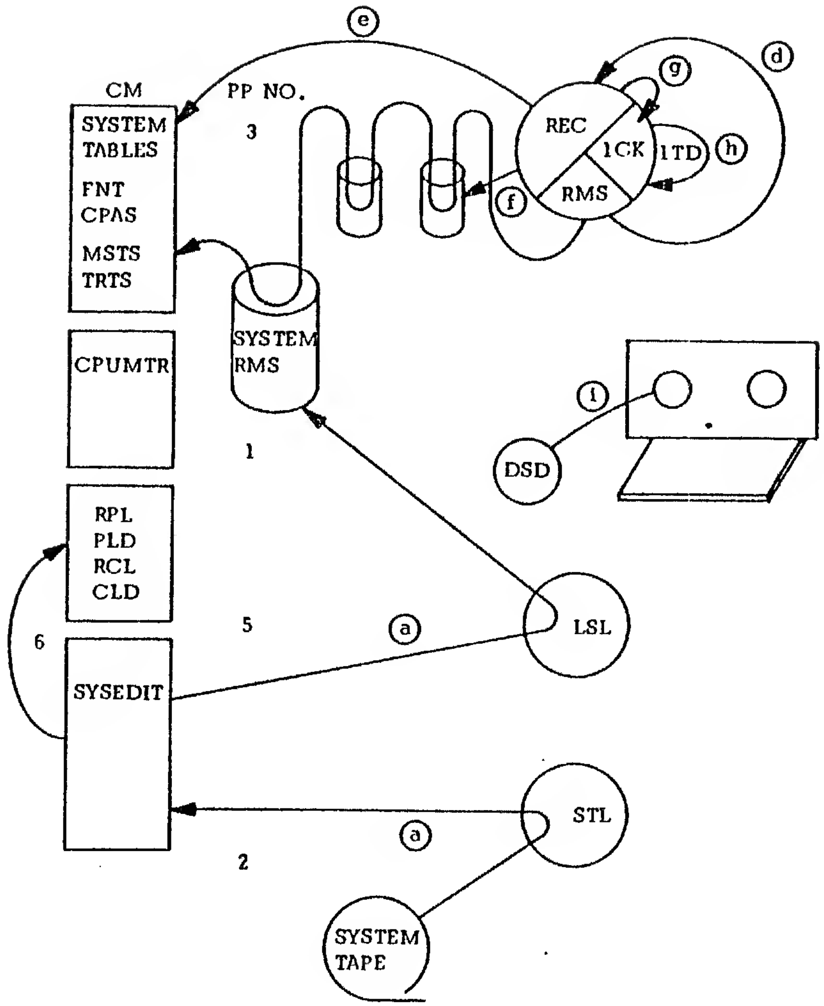
k. If 844s on system load firmware.

l. Load LSL into PPS.

m. Load SYSEDIT into CP1.

ZAP. Activate CPUMTR with an exchange jump.

Figure 24-2.1 Phase I Deadstart Startup



(a) SYSEDIT reads the SYSTEM tape and copies directly to each system RMS device.

(b) Build the RPL, PLD, RCL, and CLD via the PP routine LSL.

(c) Set MSTs from RMS labels and recover TRTs when possible and verify PF configuration.

(d) Load REC into this PP.

(e) Read SYSTEM tables and recover or create FNT, CPAs, etc.

(f) After point a has completed (SYSTEM FST byte 4 goes non-zero) recover DA files, and issue dayfile message.

(g) Load 1CK into this PP to perform a CHECKPOINT SYSTEM.

(h) Load 1TD into this PP if TELEX recovery necessary, else drop.

(i) DSD processes the IPR commands DSD,n,___:___.

Figure 24-2.2 Phase II System Activation

in length. It is preceded with a 77 table (a loader table) which is 120B bytes long. Hence the total length of PRL is 1173B. Since the load address of 6606 was specified on the deadstart panel, the last two bytes loaded into PP memory are at location 0 and 1. That is,

Word	Value
0	PRL-1 = 6725
1	garbage

When the controller senses EOR, the tape channel is disconnected and PP0 begins execution at (0)+1 or 6725+1 = 6726. Indeed, 6726 is the first executable instruction in PRL.

2) PRL checks word 13 for options. If word 13 is equal to 1, PRL inputs record two, routine 2PR, as an overlay to itself. 2PR is loaded starting in location 6606 and covers the code PRL has executed to this point. 2PR is a collection of tables and display fields.

3) PRL must connect to the operator's console to display the DS options. Normally, the console is wired to channel 10, but PP10 is connected to channel 10, therefore, PRL must release channel 10 from PP10. PRL therefore transmits across channel 10 the three bytes CON 0; LCN 0; IAM 0,0; and disconnects the channel. PP10 will input these bytes into locations 0, 1, and 2. The controller will transmit one byte of zero into location 3, i.e.,

byte
0-0
1-LCN 0
2-IAM channel 0
3-0 set by hardware OAM 3 bytes

The IAM will complete when PP0 disconnects the channel. PP10 will begin executing at location 1 which will set (A)=7777B, and hang on an IAM at channel 0. Now that channel 10 is free, PRL will connect it.

4) PRL will display the DS options and accept operator input. If any of the diagnostics are selected, PRL will input the appropriate routine (one of the next six records on the tape) into PP0 and start executing them. The operator must press the DS button to exit the diagnostics which starts the DS sequence over again. If none of the diagnostics are chosen, he hangs PP10 on CH10 with the three bytes CON 0; LCN 0; IAM 0,10, then he skips 6 records and inputs record 9, SET. There is no need to check program names since the DS tape sequence is fixed. He strips the 77 table and loads SET beginning at location 0. The first word of SET is a constant = FWA-1 of SET. Therefore, when the IAM completes, SET will begin executing in PP0.

SET initializes the system configuration by assembling system parameters such as equipment definitions and installation options from text decks on the deadstart tape. PPR and STL are read and passed to PP2. STL is started and SET prepared to input MTR.

When SET begins executing in PP0, PP10 is hung on CH10.

1) SET will transmit via CH10 an idler program to PP10 and disconnect CH10.

2) PP10 will begin executing the idler program.

3) SET will use CH10 to display the CMR/IPR DECKs and INSTRUCTIONS on the console. SET will also accept operator typeins via CH10.

4) SET uses PP10 as a CMR/IPR DECK buffer while building the appropriate DECK from operator input, if any.

5) SET communicates with PP10 via CH0. PP10 will read the appropriate CMR/IPR DECK from the DS tape via the tape channel.

6) The PP10 processors are:

processor value	routine name		
0	RSP	terminate	
1	IFP	input first buffer	from DS tape
2	ISB	input second buffer	CMR/IPR DECK
3	OFB	output first buffer	to SET via
4	OSB	output second buffer	CH0
5	ONL	output next line to SET via CH10	
6	ANL	add next line to PP10 buffer	

7) When SET has completed both CRM/IPR DECK, it will issue the RSP function to PP10 via CH0.

8) PP10 will hang itself on CH10, i.e., LCN 0; ACN 10; IAM 0,10.

The sequence is:

1) SET loads overlay CMR (next record on tape, see the system catalog), which has the processing code to make the changes to CMRDECK, reads up the text deck CMRINST (CMRDECK instructions), and reads up the specified CMRDECK.

2) CMR will display the instructions or the CMRDECK and accept input from the console which is stored in a table. When the CMRDECK is changed to the operator's satisfaction, CMR will skip all text records and load the next record, which must be ICM, as a secondary overlay.

3) ICM will build the CMR tables through the EST, when ICM completes, control is returned to SET.

4) SET loads the next record, which is IPR.

5) IPR will read IPRINST and the specified IPRDECK, display them, and accept input on the console if the last type-in from CMRDECK was NEXT. If it was GO, then IPRINST is skipped and no input is accepted. Now, IPR will set up the appropriate portions of CMR (MSCL, IPRL, etc.) and will set up all the other options specified in IPR deck. Now, IPR will return control to SET.

6) SET will skip all remaining TEXT records and will load PPR (the first non-TEXT record) into his PP buffer. He sets the location (PP buffer - 1) = PRS-1. He then (OAM) transmits the buffer starting at PP buffer - 1 into PP2 on channel two. This puts PRS-1 into location 0 of PP2. Now, SET will read the next record, STL into the same PP buffer. Then, SET will transmit the buffer to PP2. PP2 will input STL starting at location PPFW. SET then issues a DCN on channel 2. This will terminate the IAM in PP2 and execution will begin at PRS, which is the preset of PPR which then jumps to PPFW which is STL. PP0 will hang himself with an IAM on channel 0.

24.2.2 Phase II - System Activation (Figure 24-2)

STL performs the following sequence.

1) Load copy of PPR in all PPs except 0 and 1. (See paragraph 24.3) (PPR + PRS is 1077B words, thus at the end of this step each PP is IAM on location 1100B=PPFW). Also, the 1st byte transferred is PRS-1 (600) so in Step 9, control will be sent to location PRS (601) in each PP.

2) Load MTR to PP0.

3) Load and initialize CPUMTR.

4) Load DSD to PP1.

5) Load mass storage drivers to Resident Peripheral Library (RPL).

6) Load RMS to next available PP (PP3).

7) Load SLL to next available PP if system tape is to be loaded (PP4).

8) Disconnect channel 0 which will activate MTR. Now, STL will use monitor requests. He immediately issues an RPPM request for PP2, PP3, and PP4 since they are busy. (Assume that PP3 and PP4 were the next available PPs above PPR).

9) Activate all other PPs by transmitting LJM PPR and disconnecting their channels. (Note that each PP would store these two bytes at PPFW and PPFW+1).

10) If there any any 844s in the system, it loads firmware.

11) Load LSL into next available PP (PP5).

12) Load SYSEDIT from DS tape into CP1. Set up the exchange package and control point area. RA=location beyond where the CLD will be loaded and FL=rest of core.

13) Rewind DS tape.

14) Execute an exchange jump (MXN or EXN) to start CPUMTR.

15) Issues an RPPM request for PP2, PP3, and PP4 since they are busy. (Assume that PP3 and PP4 were the next available PPs above PPR).

SYSEDIT, with the help of STL to read the DS tape and LSL to write the system onto the system MS device(s) performs the following sequence.

1) Read DS tape and copy directly to each system device defined. (Used to make new deadstart tapes and for a catalog of the system file).

2) Reread the first system device as the following is performed.

3) Build the RPL in CM, stripping off all 77 tables.

4) Build a PPULIB file on each system device which is a copy of all PP routines not in the RPL stripping off all 77 tables and builds a PPULIB on any alternate devices defined while building the PLD and setting the track and sector pointers into the PPULIB file in a scratch area of central memory.

5) Eliminate the FNT/FST for the PPULIB file. Now the only record of this file is the last track and sector of the system file, and the track and sector pointers in the PLD.

6) Build the RCL stripping off the 77 tables.

7) Move the scratch copy of the PLD into CMR.

8) Build the CLD setting the track and sector pointers pointing into the system file. There is no separate copy of CM system programs created corresponding to the PPULIB.

9) Exit via ENDRUN, which sets "W", "X", and "R" status to 0.

Additional system activation requirements and functions.

1) If more than one device is designated as a system device, then:

   a) All system devices must be the same type.

   b) As SYSEDIT requests tracks, if one of the devices has an interlocked track, all other devices will not use this track. This ensures that the PLD, CLD can always point the same to all devices.

2) While the system is running (beyond deadstart), a call to SYSEDIT will:

   a) Any new or replaced CP or PP decks, libraries, etc., will be written starting at the end of the system file.

   b) A new PPULIB file is created from the system file.

c) CLD, PLD, RPL, and RCL are regenerated using any new decks added and then the system file.

d) At Deadstart Time SYSEDIT uses LIBDECK from the system file to determine which decks are CM resident and for all his directives. Each subsequent time SYSEDIT is called, a new LIBDECK is created appending the SYSEDIT directives to LIBDECK. These LIBDECKS are linked together so that SYSEDIT can recreate the PLD, CLD, RPL and RCL from any earlier time if directed by SYSEDIT (R=n).

e) The alternate library directory resides at the beginning of the PLD. This forces PP resident to check alternate libraries first, and also provides a mechanism for quickly disabling access to them.

While SYSEDIT is running in the CP, the following is accomplished in the PPs.

RMS performs mass storage recovery in the following sequence:

1) Set up MSTs from labels.

2) Recover TRTs when possible.

   a. Verify PF configuration.

3) Load REC into this PP.

REC performs system recovery in the following sequence:

1) Read system tables.

2) Recover FNT, control points, etc.

3) Wait for SYSEDIT, LSL, and STL to complete (byte 4 of system FST ≠ 0). (Step 7, page 24-6).

4) Recover direct access files (0 level only) from the disk catalog tracks and clear all DA interlocks, else from CMR.

5) Issue dayfile messages.

6) Load 1CK if checkpoint necessary (always on system load) into this PP.

7) 1CK loads 1TD if TELEX recovery necessary into this PP.

8) If 7 is unnecessary then REC will drop from the PP.

When SYSEDIT completes, LSL will issue a DPPM monitor request and jump to PPR. STL will issue an RSJM request (request scheduler) and jump to PPR.

DSD, when activated earlier by STL, will process the IPRDECK initial commands - AUTO, MAI, AB, etc.

1SJ will find that CP1 has a status of zero and will release the control point. Since there is no output file, no output or dayfile message is issued.

The system is now operational. All the DS load parameters and DS sequence descriptions are contained in the common deck COMSDSL.

24.3 PPR INITIALIZATION ROUTINE PRS

When STL sends a copy of PP resident to each pool PP, the cells IA, OA, and MA are set correctly for this PP. When STL disconnects the channel, each PP will begin executing at PRS which is the resident initialize routine. PRS is at location 601B, 600B is set zero, so that PRS will be overlayed by the 1st MS driver, PRS will:

1) Read PPCP the FWA of the PP communication area (actually IR for PP0).
2) Get (IA) which is the address of this PP's IR.
3) Subtract from it the IR of the PP0.
4) Subtract from that one PP communication region = 10B.
5) Multiply it by 21B to get the offset for the exchange package

   PP1 offset 0B
   PP2 offset 21B
   PP3 offset 42B
   PP4 offset 63B etc.

6) Read PXPP which is the address of PP1 exchange package.
7) Add PXPP to this PP offset and get the address of the EP for this PP.
8) Then he stores this address into XJ3 which is an LDC (EP address) and adds 6 to get the address of EP+6 and stores that into XJ2.
9) Determines if CEJ/MEJ exists and fixes up XJ1 accordingly.
10) Set up the rest of the direct cells.
11) Etc.
12) LJM PPFW
13) At PPFW is the instruction LJM PPR.
14) The PP is now a Pool PP.

24.4 RECOVERY

Deadstart recovery is an inhibition of part of the deadstart process. No special routines or special code is designed for different levels of recovery. The philosophy is that deadstart

is always a recovery and the levels only denote how much to recover and how much to reload. The following discussion pinpoints the process of each deadstart recovery level.

0) Permanent files are recovered if possible from the labels of the MS devices. All of CMR, all of the system, and all of the PPs are reloaded.

1) This level rebuilds the running system, all jobs, and all active files from the checkpoint file (any deadstart automatically creates a checkpoint file to start future deadstarts) instead of the deadstart tape. The FNTs are left intact. CMR is rebuilt, and all PPs are reloaded (use the CMR copy of MST for PF recovery).

2) The only difference between level 1 and 2 is that level 2 loads the system from the deadstart type. This is the method used for system test procedures.

3) This level only loads the PPs; everything else is left intact.

24.4.1 TRT Recovery

At Level 0, all track chains in the TRT denoted as PF, or reserved as flawed in the label or specified in the CMRDECK are preserved, all other tracks are set clear. No other validation is performed. Level 1 and 2 are same as level 0 except all files found in the FNT/FST are also preserved. In Level 3 no track chains are cleared.

24.4.2 More information is contained in the Installation Handbook in the Deadstart section.

24.4.3 The System Table Track is used for Level 1 and 2 recovery. It is pointed to by byte 4 of the DEVL entry of the MST for eq 0. It contains a copy of CMR and a complete copy of machine field length at CHECK POINT SYSTEM time. A partial dump is shown in Figure 24-3.

```
MST ADDRESS =        4400
00337144062400002354
77770153000153000153
00000000000000000000
00000000000000000000
42504000426100204262
70420000000000030005
15172222110000403377
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
00000000000000000000
```

DEVL byte 4 indicates that the System Table Track starts at Track 262.

Figure 24-3. System Table Track

DUMPTK(TK=252.EO=0.   DUMPTK - VER. 1 .   74/09/24. 19.48.12.   PAGE 1

WORD	TK=262 SF=0 B1=3777	B2=77	TK=262 SE=1 B1=2	B2=50	TK=252 SE=2 B1=3	B2=3
0	23312324010200010500	SYSTAB AF	00000000000000000000		70000002000004110440	# 9 DID5
1	00004252114417010000	7)I90A	90014413002400143000	A9K T LX	45000024000044110525	• T 91EJ
2	00000000411302 32514	DIXSUL	00030532000000123000	CEZ JX	45000024000144110612	• T A91FJ
3	71261405500130033201	\VLF_AXCZA	32400001134000000003	Z5 AK5 C	00000000000000000000	
4	34023040051554007076	19X5FM= #"	34004400000000006770	1 9 ^#	00000000000000000000	
5	30431277310160033405	X8J!YA_C1E	33003400330300000000	0 1 0C	00000000000000000000	
6	30071003160660400313	XGHCNF_5CK	00030531000000000000	CEY	00000000000000000000	
7	14045400707637020603	LD= #"4BFC	00030732000320410000	CG7 CP5	00000000000000000000	
10	17201430000000900002	DPL 9	00000000000000000000		00000010000004237000	H DS#
11	01060000000042634000	AF 7!5	00000000000000000000		00000012000003224000	J CR5
12	00000000040521105772	DFQH.<	00000000000000000000		00000000000000000000	
13	00000000040521106703	DFCH^C	00000000000000000000		00000000000000000000	
14	07010000040621105772	BA DFQH.<	00000000000000000000		00000000000000000000	
20	70760510301404041075	#"EHXLDDH?	00000000000000000321	CQ	00000012000014203000	J LPX
21	12010503000000003071	JAEC X\	23312324051555550000	SYSTEM	00000012000014216000	J LO_
22	34033007100316046010	1CXGHCND_H	30000030000101010100	AAAA	00000000000000000000	
23	37135400714730261074	4K= *XVH$	00000000000000000090		00000000000000000000	
24	22000017160134043111	R ONA1DYI	00000010100000100010	HH H H	00000000000000000000	
25	34063003100607333011	1FXGHFGOXT	00000000000000000000		00000000000000000000	
26	34063007100315203272	1FXGHCNPZ<	00000000004237354142	74267	00000000000000000000	
27	54007177106323002100	= \!H!S O	00000000041130232514	DIXSUL	00000000000000000000	
30	54007176300619752100	= \"YFH?O	55344457353457343557	19.21.12.	00000012000003205000	J CP/
31	66205010300612033414	!P_HXFJC1L	55423750334450353757	74/09/24.	00000000000000000000	
32	50140010340637040563	/L H1F4DE!	55151722221105235520	MORRIES P	00000000000000000000	
33	30053201340530070100	XEZA1EXGA	05222317160114551322	ERSONAL KR	00000000000000000000	
34	70030100721754007231	#CA <D= <Y	17161723553557345700	ONOS 2.1.	00000000000000000000	
35	14056001300332013402	LF_AXC7A19	00000000000000000000		00000000000000000000	
36	20007230310160033405	P <XYA_C1E	00000000000000000000		00000000000000000000	
37	30071003160661707321	XGHCNF(#>O	00000000000000000000		00000000000000000000	
40	50007324137733105400	/ >TK!0H=	00000000000000010001	A A	20000003002064247000	P C P'T#
41	73240306370207443101	>TCF4BG9YA	40000000000000000000	5	20000111004064240000	P AT 5'T
42	60033405300310711251	_C1EXCH\J(	00000000000000020003	B C	00000000000000000000	
43	11500566300619960753	I/E!XFHFG!	00011200000000020002	AJ 9 B	00000000000000000000	
44	30071003160560103010	XGHCNE_HXH	00000000000000000000		00000000000000000000	
45	12340554300710031605	J1F=XGHCNF	00000000000000000000		00000000000000000000	
46	60101400340450047321	_HL 10/C>O	00030532000000123000	CE7 JX	00000000000000000000	
47	53040010054136041104	$D HE63DID	00000000000000003020	XP	00000000000000000000	
50	05703005320134053007	E#XE7A1EXG	00000000000000227704	R!D	00000013000015245000	K MT/
51	01007217000000000000	A <O	00000000000000000000		00000013000015245001	K MT/A
52	00000000010073262000	A >VP	00000000000000000000		00000000000000000000	
53	11130200131450001100	1KB KL/ I	00000000000000000000		00000000000000000000	
54	05100700120316020320	FHA JCNBCP	00000000000000000000		00000000000000000000	
55	14020100131002006305	LBA KHB !E	00000000000000000000		00000000000000000000	
56	04720200635004061400	B<B !/DFL	00000000000000000000		00000000000000000000	
57	34571444010013103057	1.LBA KHX.	77770112000000000000	!!AJ	00000000000000000000	
60	60201701501055001312	_POA_H, KJ	00000000000000013667	A3^	00000000000000000000	
61	30131204055130701277	XKJDE(XPJ!	77770000000000000000	!!	00000000000000000000	
62	34055100055160103010	1E(E(_HXH	00000000000000013707	A4G	00000013000016245002	K NT/3
63	10060704140601001310	HFGOLFA KH	00012605000000000000	AVE	00000013000016245003	K NT/C
64	02000547301410031620	B E*XLHGNP	51701073010010737773	(#H>A H>!>	00000000000000000000	
65	32725400460610632300	7<= -FHIS	00002007031261131073	PCCJ(KH>	00000000000000000000	
66	21005400450530220504	0 = -EXRED	00000000000000000000		00000000000000000000	
67	14040100131030213406	LDA KHYD1F	00000000000000000000		00000000000000000000	
70	02004577300234273003	B *!XB1WXC	00000000000000000000		00000020000023030001	P SC A
71	17013430060135303727	0A1XFC3X4H	00000000000000000000		00000020000023030002	P SC 3
72	30713422140134231024	X01BLA1SXT	00000000000000000000		00000020000023030005	P SC E

Figure 24-3. System Table Track (Continued)

```
          DUMPTK(TK=262,EQ=0.                                                     DUMPTK - VER. 1      74/09/24. 19.48.12.   PAGE   2
 73       13771105342401007326  K:IEITA >V    00000000000000000000                         00000000000000000000
 74       00000000000006365322       F3:R     00000000000000000000                         00000000000000000000
 75       03150010640443001063  CM H'D8 HI    00000000000000000000                         00000000000000000000
 76       67545055200166466300  ^=/ PAF-:     04000000760000000000  D    "                 00000000000024050003        TE
 77       43400043000106410677  95 8 AF6F:    04000000770000000000  D    :                 20000000000016050000  P     NE

WORD      TK=262  SE=3    B1=4    B2=100      TK=262  SE=4    B1=5     B2=100              TK=262  SE=5    B1=6     B2=100
 0        23312324051500010700  SYSTEM AG     00000000000000000000                         00000000000000000000
 1        00004001400100010005    5A5A A E    00000000000000000000                         00000000000000000000
 2        26011411042530000500  VALIDUX E     00000000000000000000                         00000000000000000000
 3        00000000000000000001           A    00000000000000000000                         00000000000000000000
 4        23011426012205000700  SALVARE G     00000000000000000000                         00000000000000000000
 5        00000000000000000001           A    00000000000000000000                         00000000000000000000
 6        22052305300406000500  RESEXOF F     00000000000000000000                         00000000000000000000
 7        00000000000000000001           A    00000000000000000000                         00000000000000000000
 10       22052305302606000500  RESEXVF E     00000000000000000000                         00000000000000000000
 11       00000000000000000001           A    00000000000000000000                         00000000000000000000

WORD      TK=262  SE=11   B1=12   B2=100      TK=262  SE=12   B1=13    B2=100              TK=262  SE=13   B1=14   B2=0      EOR
 0        00000000000000000000                00000000000000000000                         00400657062400002315  5F.FT  S9
 1        00000000000000000000                00000000000000000000                         77770153000153000153  :;A$ A$ A$
 4        00000000000000000000                00000000000000000000                         42604000426100204262  7_5 7( P7)
 5        00000000000000000000                00000000000000000000                         70420000000000000005  #7       E
 6        00000000000000000000                00000000000000000000                         15172222110000400377  MORRI  SD:
 20       00000000000000000000                00000000000000000000                         42374002400340044017  745D5C5D5 D
 21       00000000000000000000                00000000000000000000                         40054006400740100017  5E5F5G5H D
 22       00000000000000000000                00000000000000000000                         40114012401340140017  5I5J5K5L D
 23       00000000000000000000                00000000000000000000                         40154016401740200017  5M5N5O5P D
 24       00000000000000000000                00000000000000000000                         40214022402340240017  5Q5R5S5T D
 25       00000000000000000000                00000000000000000000                         40254026402740300017  5U5V5W5X D
 26       00000000000000000000                00000000000000000000                         40314032403340340017  5Y5Z5051 D
 27       00000000000000000000                00000000000000000000                         40354036403740400017  52535455 D
 30       00000000000000000000                00000000000000000000                         40414042404340440017  56575859 D
 31       00000000000000000000                00000000000000000000                         40454046404740500017  5+5-5*5/ D
 32       00000000000000000000                00000000000000000000                         40514052405340540017  5(5)5$5= D
 33       00000000000000000000                00000000000000000000                         40554056405740600017  5 5,5.5_ D
 34       00000000000000000000                00000000000000000000                         40614062406340640017  5(5)5:5' D
 35       00000000000000000000                00000000000000000000                         40654066406740700017  5a5!5^5# D
 36       00000000000000000000                00000000000000000000                         40714072407340740017  5\5<5>5& D
 37       00000000000000000000                00000000000000000000                         40754076407741000017  5?5"5;6  D
 40       00000000000000000000                00000000000000000000                         41014102410341040017  6A6B6C6D D
 41       00000000000000000000                00000000000000000000                         41054106410741100017  6E6F6G6H D
 42       00000000000000000000                00000000000000000000                         41114112411341140017  6I6J6K6L D
 43       00000000000000000000                00000000000000000000                         41154116411741200017  6M6N6O6P D
 44       00000000000000000000                00000000000000000000                         41214122412341240017  6Q6R6S6T D
 45       00000000000000000000                00000000000000000000                         41254126412741300017  6U6V6W6X D
 46       00000000000000000000                00000000000000000000                         41314132413341340017  6Y6Z6061 D
 47       00000000000000000000                00000000000000000000                         41354136413741400017  62636465 D
 50       00000000000000000000                00000000000000000000                         41414142414341440017  66676869 D
 51       00000000000000000000                00000000000000000000                         41454146414741500017  6+6-6*6/ D
 52       00000000000000000000                00000000000000000000                         41514152415341540017  6(6)6$6= D
 53       00000000000000000000                00000000000000000000                         41554156415741600017  6 6,6.6_ D
 54       00000000000000000000                00000000000000000000                         41614162416341640017  6(6)6:6' D
 55       00000000000000000000                00000000000000000000                         41654166416741700017  6a6!6^6# D
 56       00000000000000000000                00000000000000000000                         41714172417341740017  6\6<6>6& D
 57       00000000000000000000                00000000000000000000                         41754176417742000017  6?6"6;7  D
 60       00000000000000000000                00000000000000000000                         42014202420342040017  7A7B7C7D D
 61       00000000000000000000                00000000000000000000                         42054206420742100017  7E7F7G7H D
```

Figure 24-3. System Table Track (Continued)

J TK=262 SF=151 B1=152 B2=100 TK=262 SF=152 B1=4271 B2=100 —— Indicates Link to track 271

DUMPTK(TK=262,EQ=0. DUMPTK - VER. 1 74/09/24. 19.48.12. PAGE 38

0	14321006335134513075	L7HF0((1(X?	00000100152550001351	A NU/ KI
1	62500100010314023401	)/A ACL*1A	34073203067150070001	1G7GF\/G A
2	1444F30113210100 1141	L9/AKDA 15	34112000242334025300	11P TS19$
3	20000106503030743407	P AF_XX&1G	13510469500200013311	K(O_/8 A01
4	03127000777632210706	CJP 1"7OGF	05114007430204111402	E15G8001L8
5	340130071070540113 21	1AXGH#=AKD	35025300135105641405	29$ K(E'LE
6	20777577350704461621	P!7!2GD-N)	35070346500700026020	2CG-/C B_P
7	60103010047030131277	_HXHD#XKJ!	17016040500700043424	DA_5/G D1T
10	34011705065330071622	1AOEF1XGNR	14013422207774773544	LA1RP1&!29
11	60201501601030212177	_PNA_HXON!	50070002622017016240	/C 9)POA)5
12	00120641301031110447	JF6XHY1D*	01001561010017105000	A N(A OH/
13	30303210074410143131	XX7HG9HLYY	13310573144360103011	KYE>L8_NXI
14	32110740302121777677	Z1G5XOO!":	10130717200001116010	HKGOP AI_N
15	07672177011706453012	G^O!AOF+XJ	20000106603030113230	P AF_XX1ZX
16	22003700330110140200	R 4 OAHLB	10143112373106030100	HLYJZYFCA
17	22373011540012663221	R4X1= J!7)	20051405602037226040	PELE_P4R_S
20	07113001110305040200	G1XAICEOB	32200742304010060671	7PG7X5HFF\
21	21130503010011623007	OKECA 1)XG	10041056550016123043	HOH, NJX8
22	10703411143702000354	H#11L49 C'	10060762304410031602	HFG)X9HCNR
23	30110533200012653411	X1EOP J01I	50031603601030101015	_CNC_HXHHM
24	14013412141607000354	LA1JLNB C'	05033013041330050445	FCXKDKXEO+
25	30011103051202002113	XA1CEJR OK	20352320020005330200	P2SPB EO9
26	04073007107034111415	DGXGH#11LM	22640100174220000111	R'A O7P A1
27	07000364307410703411	9 C'X&H#11	63101474020020572000	_HL&R P.P
30	14372200035401001162	L4B C'A I)	01116210200020220100	A1)HP PRA
31	00000000000000000000		23273404230000320000	PH1)S Z
32	00000000000000000000		77665400205334011441	!!= P$1AL6
33	14150700044601001336	LMD)-A K3	60103010722003777341 0	_HXHR 4!1N
34	14023411143207000364	L911LZ9 C'	14416210145402000364	L6)HL=B C'
35	30110406307761142423	X1DFX!(LTS	30741070550100013075	X&N# A AX?
36	30111001550013513433	X1H8 K(1C	63702057010001030100	!#P)A ACA
37	14046030141507000437	LO_XLMD O4	70553131341210633130	P,YY1JH!YX
40	14023530601033310442	LR2X_HOYD7	34110370010070661440	1IC#A P1LS
41	30100471301413401014	XHD\XLKSHL	60101601602020000106	_HNA_PP AF
42	17050654160534025002	OEF'NE1P/B	60303014050214203131	_XXLEDLPYY
43	15203401301312771705	HP1AXKJ:OE	34741063313034231441	1TH!YX1SL6
44	06531605101433010700	F$NFHLOA9	62200354010021123007	)PC=A OJXG
45	22373030160160203002	P4XXNA_PXB	16206010301404713013	NP_HXLD\XK
46	11040420301416000513	10OPYLA EK	34021401340336031164	19LA1C3CI'
47	30243211071030243212	XT71GHXI7J	04523002100631036012	D)X9HFYC_J
50	06053624303016016220	FE3TXXNA)P	30120454301512771014	XJD=XMJ!HL
51	01001364302010714403	A K'XPH\9C	33160462340010143102	ONO)1 HLYP
52	11770510372405643722	1:FH4TE'4P	10056010160160403013	HF_HNA_5XK
53	07301500342403573724	GXM 1TC.4T	12773114050530433415	J!YLEEX81M
54	04233022540300014103	OSXP=C A6C	30443416301512371005	X91NXHJ4NF
55	04503030160154030002	D/XXNA=C 9	31021006311660401602	Y9HFYN_5NB
56	30245403000330125403	XT=C CXI=C	60121601601030402300	_JNA_HX5S
57	00041405350303143030	DLE7CCLXX	17250523304123002420	OUESX6S 1P
60	60402077747735443030	_5P!&!29XX	05173042230025240513	EOX7S UTEK
61	62401400342230123424	)5L 1RXJ1T	30431377051030133315	X8K!EHXKOM
62	01001436000000010002	A L3 A B	10143114331601002112	HLYLONA QJ
63	00070001010015251434	B AA MULO	30000403010021253002	X OCA OUXB
64	60301400340134023033	_XL 1A1RXO	10063103601234000100	HFYC_J1 A
65	10113101100331343102	H1YAHCY1Y9	21620100223634001066	O)A R31 H!
66	60103614321307031400	_H3L7KGCL	35001434601030131014	Z LO_HXKHL
67	34143033101131011003	1LXOH1YAHC	31143100601003530000	YLY _NC!
70	31343102671036021103	Y1Y9)H3P1C	34020100107300000151	19A H> A(
71	05523402360111050546	E119JAIEE-	07007037200020163401	B P4P PN1A

Figure 24-3. System Table Track (Continued)

DISPLAY ROUTINES DSD AND DIS 25

25.0 INTRODUCTION

DSD and DIS are display routines which require a dedicated PP. DSD is placed in PP1 by STL during deadstart (see Section 24, Deadstart) and remains there for the duration. DIS is called to a pool PP by the operator command X DIS. When DIS is in a PP, it gets a control point and retains both the PP and the control point until it is ended by a DROP or n.DROP command. DSD and DIS (and any other PP routine, (such as snoopy, LEM, etc., which desires the operator console channel) will toggle the display console by the use of the "*" key. User information is contained in the Installation Handbook, Instant Manual, and Operator's Guide.

DSD and DIS use the common decks: COMDSYS, COMDDIS, COMDDSP, COMDTFN. Any routine wishing to use the console should use these common decks and the macros in DSD and DIS. (A complete listing of CALLDIS is obtainable within the discussion in Section 20.)

25.1 DSD DISPLAY ROUTINE

DSD runs independent of a control point, however, when an input requires operation on a control point area (change memory location, n.drop, etc.), DSD will take on the attribute of being assigned to the control point in question until the operation is complete, usually by calling 1DS. (See Figure 25-1.)

DSD is responsible for all the system displays, accepting all system keyboard input, and initiating all action desired from this input.

If a typein requires some control card action, it calls 1DS to initiate this action. If a typein specifies a particular display, DSD will load the appropriate overlay to fill the screen buffers.

As DSD receives input, it processes them one character at a time as they are received. Checking is performed on each character to validate the entry. DSD checks the first character and loads the syntax routines necessary to process any command which begins with this character (9AW, 9AX, 9AY, 9AZ, 9AO). As the typein's continue, the entry is narrowed down to a unique entry, at which time the remainder of the entry is filled in by the input processor. At this point, the entry is considered complete, and the keyboard echo line is flashed to indicate the complete entry (This syntax facility can be toggled on and off by the typein 99.)

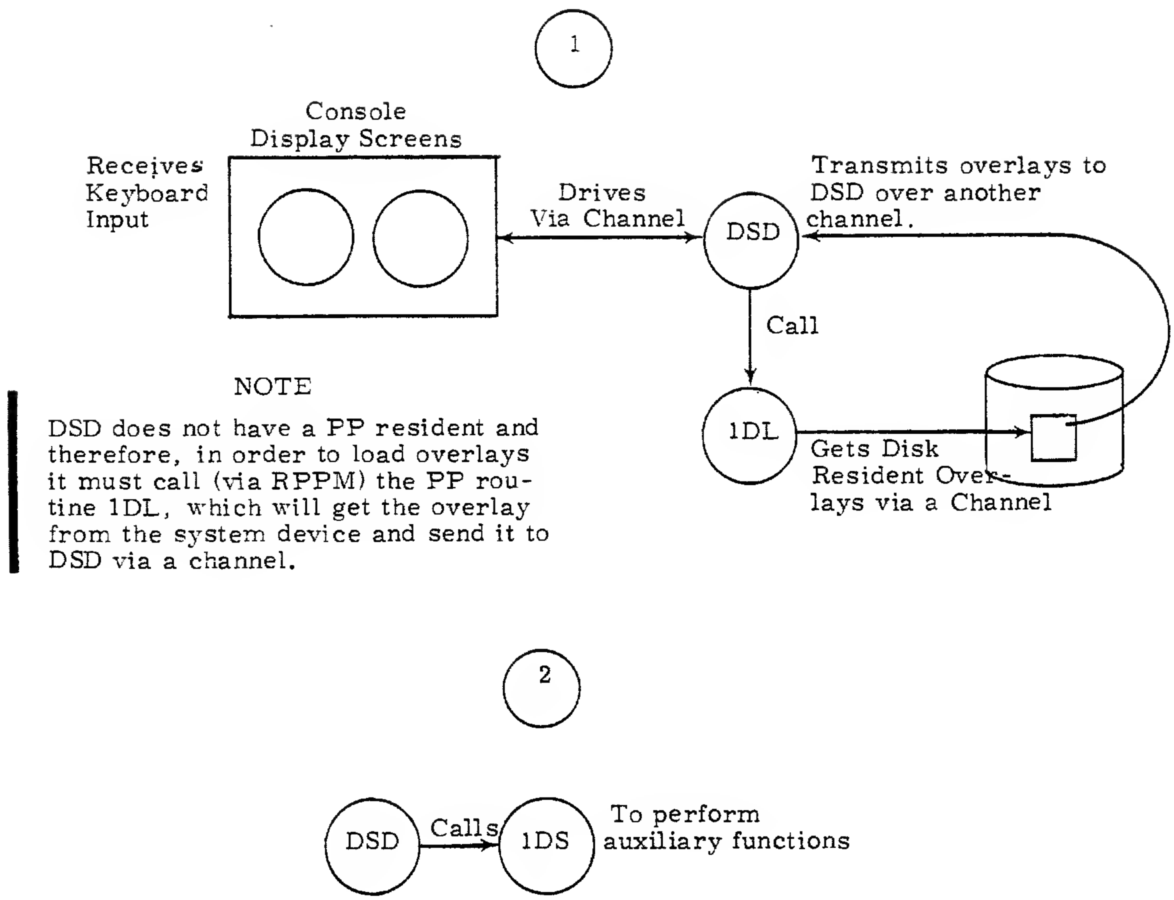


Figure 25-1. DSD Overview.

Each screen display is controlled by a separate overlay. All these DSD overlays can be seen in a system catalog. The overlays set up the buffer, and the main loop displays it

An analyst wishing to define an overlay should get a listing of one of the DSD overlays, and modify a copy for his own needs. The analyst should use the macros defined in DSD for defining:

1. Overlays (use OVLN macro)
2. Overlay entry points (use ENTRY macro)
3. Command processing (use ENTER macro)

Listings of these macros are provided subsequently.

DSD uses three types of overlays - SYNTAX, DISPLAY, and COMMAND. The following macros facilitate the organization and communication between overlays and the main program.

Overlays may reside in "RPL" or "PLD". For those overlays which reside in "PLD", DSD calls the program (1DL) to process the physical load of the overlays, since there is no copy of PLL or any PP resident type load routines. 1DL will transmit the overlay to DSD via a channel.

SYNTAX and COMMAND overlays are absolutely originned. Display overlays are written as location free routines since two must reside in DSD for the two display screens.

1. ENTRY - Define overlay entry point.
   ENTRY NAME, D
   ENTRY NAME = Name of entry point
   (D) if present, defines display overlay entry
   Point is the current value of the origin counter.
   EXIT NAME = (address) + last two characters of overlay name

2. OVLN - Generate overlay name.
   OVLN
   EXIT (N.) = last two characters of overlay name.

3. DISPLA - Display data. (See Section 19 on K Display programming for description of the screen matrix.)
   DISPLA X, Y, (TEXT)
   Entry X = X-coordinate
   Y = Y-coordinate
   TEXT = display text

4. If coordinates are not present, display text at current position.

```
DISPLA  MACRO X,Y,T
        LOCAL I,J,K
        QUAL
K       MICRO 1,,$T$
.1      MICCNT K
I       SET    .1+1
I       SET I/2
        LDN   K
        OAM   J,CH
        QUAL *
DIS     RMT
        QUAL
J       BSS    0
        IFNE   X,,2
        CON    X
I       SET    I+1
        IFNE   Y,,2
        CON    Y
I       SET    I+1
        DATA   HS≠K≠S
K       EQU    I
        QUAL   *
DIS     RMT
        ENDM
```

5. ENTER - Specify command entry

The "ENTER" macro is used for specifying the format of the keyboard commands.

```
NAME    ENTER COMMAND, LOCK
        ENTER NAME = address of command processing routine as specified
                     by "ENTER" macro.
              COMMAND = SYNTAX for keyboard entry.
              LOCK (if present) = SYNTAX is under lock control.
```

NOTE

Special fields may be specified by the following characters.

↑ 70 (11-8-5) alphanumeric field.

↓ 71 (11-8-6) octal field.

< 72 (12-0) used as XY - any character between *X* and *Y* may be used in this field.

> 73 (11-8-7) used as XYZ - any character in the set *XYZ* may be used in this field.

≤ 74 (8-5) terminate scan - characters in any format may follow.

≥ 75 (12-8-5) set new SYNTAX table - SYNTAX field described by (address) will be used for remainder of fields.

Examples of some ENTER macro use are listed in Table 25-1.

TABLE 25-1. Macro ENTER

Name	Macro	Entry	Name	Macro	Entry
CDS	ENTER	(≥≥ .)*1	IAN	ENTER	(IAN↓.), LOCK
DCC	ENTER	(>ACDFGHJKL>,↓.)*2	AUT	ENTER	(MAINTENANCE.)
DFC	ENTER	(>CDFG>06,↓ .)*3	MCH	ENTER	(MCH↓.), LOCK
ACD	ENTER	(ACCOUNT,↓ .)	OAN	ENTER	(OAN ↓.), LOCK
ACN	ENTER	(ACN↓.), LOCK	OFE	ENTER	(OFF ↓.)
AUT	ENTER	(AUTO.)	ONE	ENTER	(ON↓.)
RSA	ENTER	(A.)	QUE	ENTER	(QUEUE≤)
IDL	ENTER	(BLITZ.)	STM	ENTER	(STEP.), LOCK
DCH	ENTER	(DCH↓.), LOCK	STP	ENTER	(STEP, ↓.), LOCK
DCN	ENTER	(DCN↓.), LOCK*4	SYG	ENTER	(SYSGO.)
SEN	ENTER	(EN;)	TIM	ENTER	(TIME ;), LOCK
ELD	ENTER	(ERRLOG,↓)	ULK	ENTER	(UNLOCK.)
			UNS	ENTER	(UNSTEP.), LOCK
FCN	ENTER	(FCN↓.), LOCK			
FCN	ENTER	(FCN↓,↓.), LOCK	IIF	ENTER	(X.< AZ↑ .)*6
			IIF	ENTER	(X< AZ↑, ↓)*7
IDL	ENTER	(IDLE.)*5			

*1 - means accept two legal display names such as AB, or CE, or AN or KB, etc.

*2 - means accept octal field such as A, 5 or C, 5 for specific CP number display.

*3 - C4, octal field or C0, C1, C2, C3, for specific read of CM.

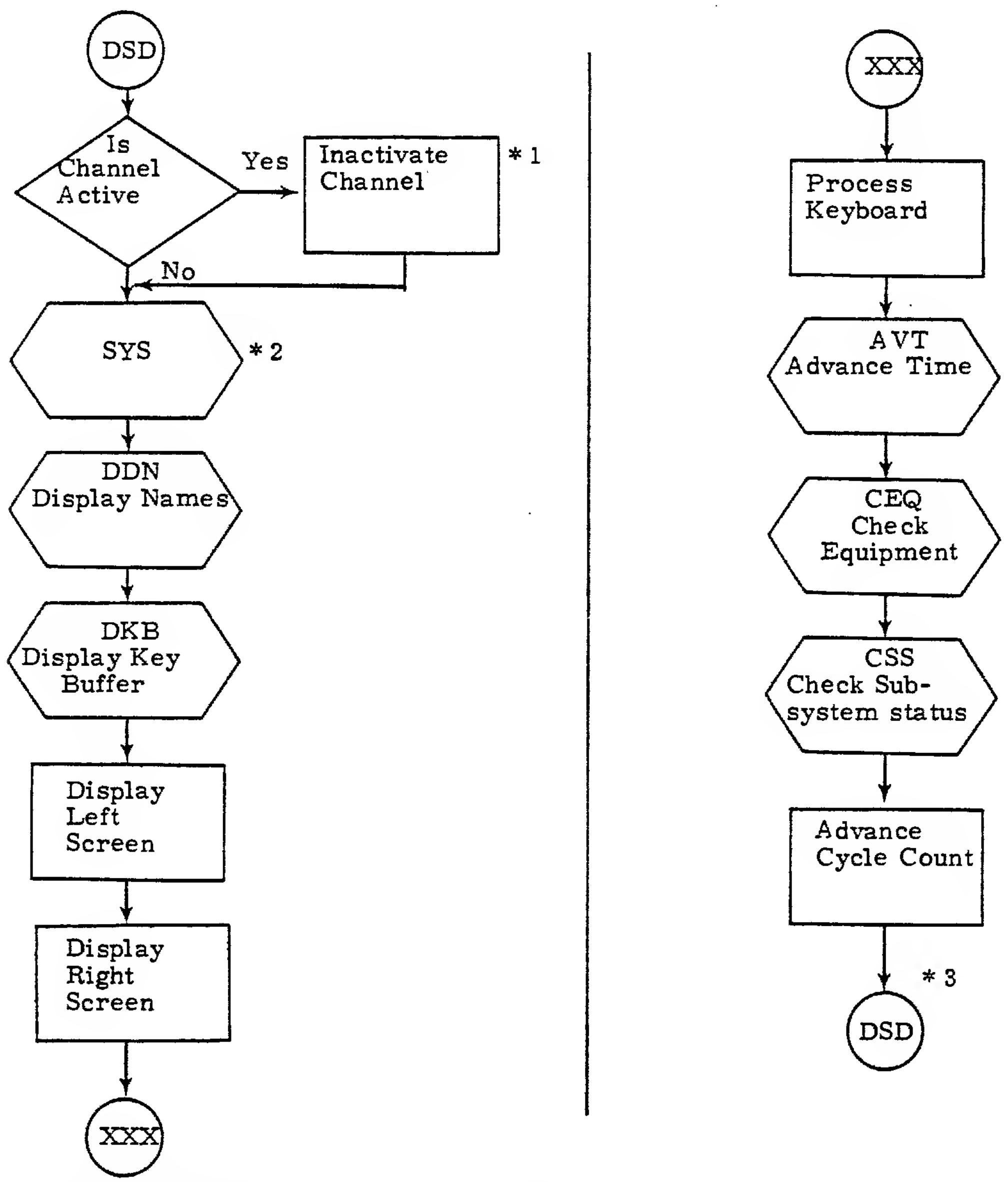
*4 - Can type in DCN only if console unlocked.

*5 - Can type in IDLE in LOCK or UNLOCK status.

*6 - Can type in X. any sequence of Alphas followed by numeric field.

*7 - Same as *6 plus "," followed by any octal field.

Through the discreet use of these macros and reading the DSD code, an analyst can define his own overlays. All the input and output processing can make the use of the features present in the DSD code. A flowchart of the main loop is presented in Figure 25-2.



* 1 Ensure channel is free before attempting any action on it.
* 2 COMDSYS display system status.
* 3 Loop. Screen must be refreshed 50 times a second

Figure 25-2. DSD Main Loop

25.2 1DS

1DS processes those functions that DSD cannot process. 1DS is also called to enter jobs for IAJ in certain cases.

The 1DS call is:

Direct Cells IR =	IR		IR + 1		IR + 2		IR + 3	IR + 4
	1DS			SC	JC	REQ	PARAMS	ORDINAL

where:

SC = System control point

JC = 0, control point to perform at

REQ = Request number

PARAMS = Parameters

ORDINAL = Job ordinal if JC≠0. (FNT address)

Table 25-2 is a list of all the request processors, while Table 25-3 lists the value of JC, REQ, PARAMS, and ORDINAL for each processor

1DS is called by DSD to process the functions in Table 25-2. If JC is specified, 1DS gets that control point and then processes the request. (See Figure 25-3).

TABLE 25-2. TABLE OF REQUEST PROCESSORS — IROP

TRQP	Request Number*[1]	Processor Name*[2]	Description
	0	DSF	Load display buffer
	1	MSG	Send dayfile message
	2	GGO	Go
	3	ONS	On Switch
	4	OFS	Off Switch
	5	ECB	Enter central buffer
	6	PGF	Purge files
TRQP1	7	RRJ	Rerun job
	10	ITJ	Initiate jobs from table
	11	IJC	Initiate job call
	12	JFD	Dayfile dump
	13	ACD	Account file dump
	14	ELD	Error log dump
	15	LOD	Load input jobs
	16	DPQ	Dump print queue

TABLE 25-2 (Cont'd)

TRQP	Request Number* 1	Processor Name* 2	Description
	17	ICJ	Initiate control card job
	20	MES	Issue TELEX message
	21	WAR	Issue TELEX warning message
	22	DIA	Send TELEX user a message
	23	CFO	Enter data to running job
	24	ROL	Rollout job
	25	SJP	Enter job CPU priority
	26	SJP	Enter job queue priority
	27	STL	Set job time limit
	30	AEJ	Assign equipment to job
	31	DIS	Call DIS to job
	32	ISS	Initiate specified subsystem
	33	IAS	Initiate all enabled subsystems
	34	BIO	Process BATCHIO operator commands
	35	EUF	Enter MAGNET UDT field
	36	TPS	Toggle PF status
TRQPL	37		End-of-table

*1 Entry = 1 word indexed by request number

*2 Processor name = Address of request processor

Functions that require interlock cleared (i.e., UNLOCK status are) 1, 5, 11, 17, 20, 21, 22, 23, 0.

25.3 DIS

DIS is always associated with a control point, while DSD is seldom so associated. (See Figure 25-3).

When the operator types in X.DIS, DSD will call 1DS which will get a control point and call DIS to this control point. A control point can also request DIS via a control card call.

DIS will check that the user is validated for DIS if the system is in DEBUG status. If DIS is a direct call, no validation is necessary.

DIS controls the displays in much the same manner as DSD. In addition, control point information exchange packages, breakpoint, 026, and control card calls can all be initated via DIS.

Table 25-3. 1DS Processor Values

Name	REQ	JC*	PARAMS = IR + 3	ORDINAL = IR + 4
DSF	0		≠0 Back space file	FNT address
MSG	1		FWA message	
GGO	2			
ONS	3		Switch Number	
OFS	4		Switch Number	
ECB	5		Address of message buffer	
PGF	6		File type if PURGEALL	FNT address if 1 requested
RRJ	7		Rerun priority	
ITJ	10		Table number	
IJC	11		Address of job name	Field length
DFD	12		Equipment number	
ACD	13		Equipment number	
ELD	14		Equipment number	
LOD	15		Equipment number	ID on FNT
DPQ	16		Equipment number	ID on FNT
ICJ	17		Address of job name	Field length
MES	20		Address of message	
WAR	21		Address of message	
DIA	22		Address of message	Terminal number
CFO	23		FWA of message	
ROL	24		Rollout time, 0 if not timed	
SJP	25		Priority	
SJP	26		Priority	
STL	27		New time limit	
AEJ	30		Equipment	
DIS	31			
ISS	32		Desired control point	Queue priority
IAS	33			
BIO	34		Parameters	BATCHIO flag and buffer pointer number
EUF	35		Address of entry	
TPS	36		Bit to toggle	Equipment number

* JC may or may not be set depending on the circumstances of the call.

In general, DIS provides a convenient means to alter the running of a job, or (if called to a blank control point), initiate the operation of utility programs.

All displays are displayed from a buffer by the main loop. The buffers are filled and modified by an overlay routine as in DSD. An analyst wishing to add or modify a display should follow the same procedure as in DSD.

DIS is a transient routine that may reside in any PP and will remain for the duration of the job

The DIS call is:

IR =	DIS	D	CP	0	Display Console Equipment

where:

D = 1, direct call to control point.

Figure 25-4 is a flowchart of the main loop. A list of the DIS overlays is given in a system catalog.

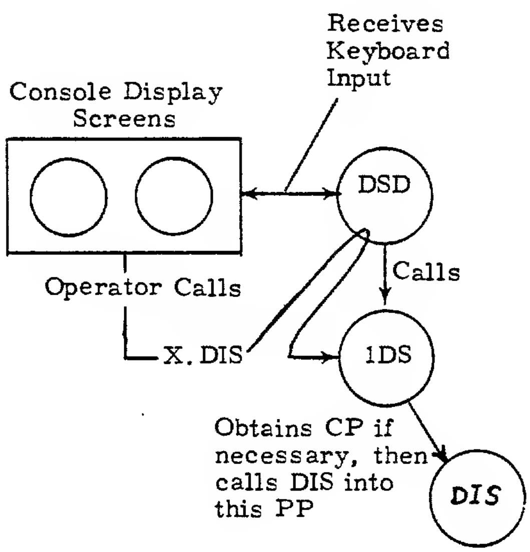
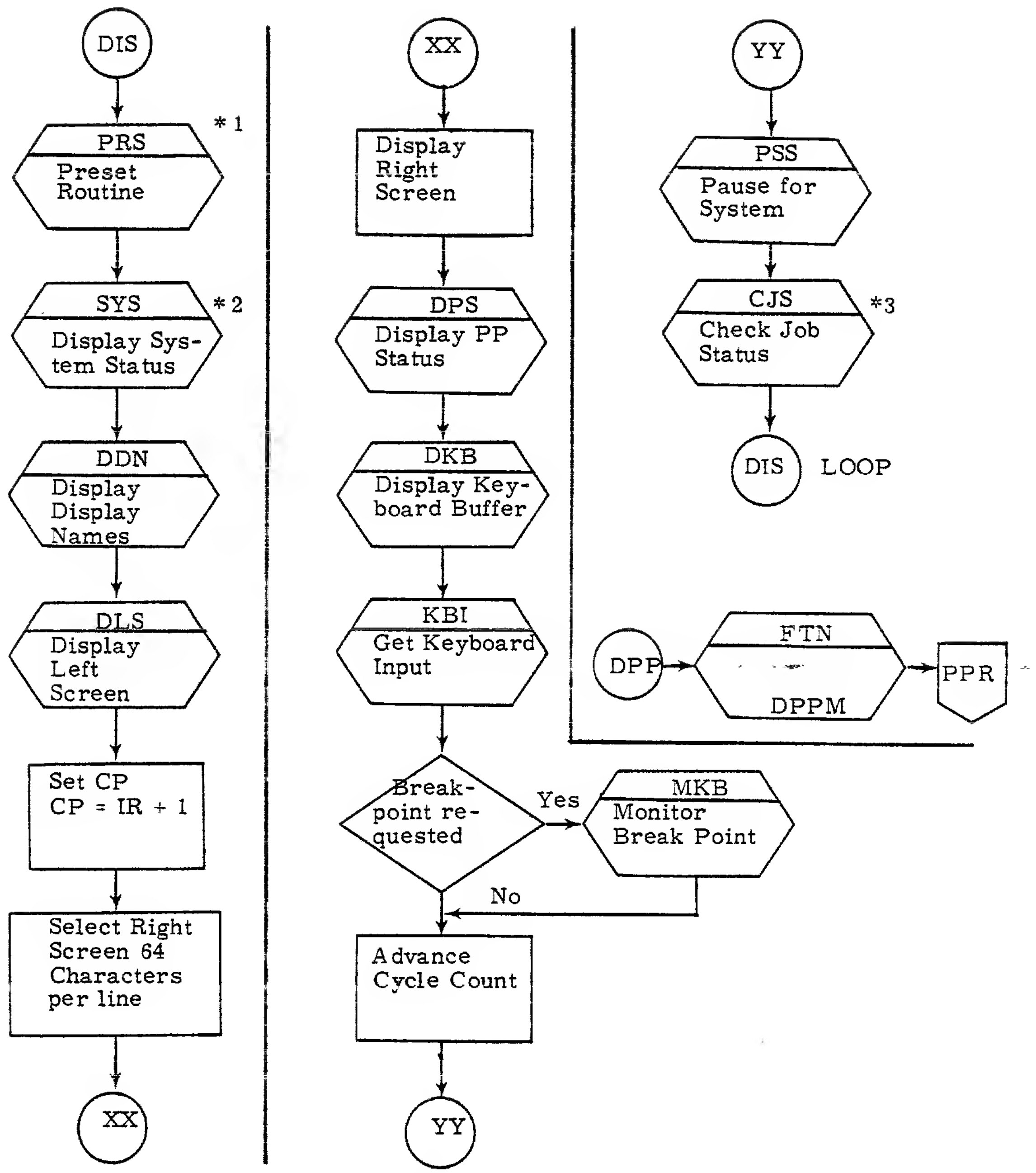


Figure 25-3. DIS Overview.



* 1 presets storage, determines if this is a legal call.

* 2 left screen

* 3 if type in DROP, or error flag set with no reprieve, and no activity, then call 1AJ to a different PP and jump to DPP.

Figure 25-4. DIS - Main Loop

26.0 INTRODUCTION

KRONOS 2.1 supports most of the SCOPE 3.4 product sets.

6RM - SCOPE record manager
STS - Status processor
EMG - SDA/SIS message generator
RPV - Reprieve central program
PFE - Extend/Alter function
ACE - Advance control card
PRM - Permission checking function
CKP - Checkpoint restart
REQ - Request equipment assignment
DMP - Dump field length
FORTRAN - FTN and RUN.
COBOL
SORT

6RM is extensively described in the 6RM-Record Manager Reference Manual Several examples of 6RM used by FORTRAN, COBOL, and SORT are shown in Figure 26-1. One very important fact: All of the libraries SYSIO, FORTRAN, COBOL, SORT, etc. must be at the same PSR level. They are very interdependent and catastrophic results will occur if libraries are at different levels.

26.1 6RM

- User sets up a FIT
- 6RM is a group of object time routines which use the FIT and generate a call to CIO.

6RM accepts:

File	ORG
SQ	Sequential
IS	Indexed sequential
DA	Direct Access
WA	Word Addressable

```
                AO 0016                                   COBOL    V40P348    73/09/07. 11.27.24.    PAGE    1

          IDENTIFICATION DIVISION.
          PROGRAM-ID.  SAMPLE.
          AUTHOR.  DDT.
          ENVIRONMENT DIVISION.
          CONFIGURATION SECTION.
          SOURCE-COMPUTER.  6600.
          OBJECT-COMPUTER.  6600.
          INPUT-OUTPUT SECTION.
          FILE-CONTROL.
              SELECT TEST1 ASSIGN TO TEST.
          DATA DIVISION.
          FILE SECTION.
          FD  TEST1
              LABEL RECORDS ARE OMITTED
              DATA RECORD IS RECOUT
              FILE CONTAINS ABOUT 100 RECORDS
              BLOCK CONTAINS 640 CHARACTERS
              RECORD CONTAINS 100 CHARACTERS.
          01  RECOUT  PIC X(100).
          WORKING-STORAGE SECTION.
          77  COUNTER PIC 9(10)  VALUE IS 1.
          01  REC-IMAGE.
              02 HDR  PIC X(15) VALUE IS # 6RM CHECK     #.
              02 DYNAMIC PIC X(10).
              02  FILLER PIC X(10) VALUE IS SPACES.
              02 TLR PIC X(20) VALUE IS # FILE INTERCHANGE   #.
              02 FILLER PIC X(45) VALUE IS SPACES.
```

COBOL/FTN

File Interchange

"C"

"F"

BLOCKING Records

```
        SAMPLE          AO 0016

          PROCEDURE DIVISION.
          START.
              OPEN OUTPUT TEST1.
          AGAIN.
              MOVE COUNTER TO DYNAMIC.
              MOVE REC-IMAGE TO RECOUT.
              WRITE RECOUT.
              ADD 1 TO COUNTER.
              IF COUNTER GREATER THAN 100 GO TO HALT.
              GO TO AGAIN.
          HALT.
              CLOSE TEST1.
              STOP RUN.
SAMPLE  LENGTH IS  000077
 053C00B SCM USED
```

Figure 26-1 6RM Examples

```
PROGRAM      SAMP                                   CDC 6600 FTN V4.0+P348 OPT=1  73/09/07. 11.27.27.      PAGE     1

                      PROGRAM SAMP (TEST,TAPE1=TEST,OUTPUT)
                      DIMENSION BUFFER(10)
                      DO 100 J=1,100
                      READ (1,10) (BUFFER(I), I = 1,10)
                 10   FORMAT (10A10)
                      PRINT 20,BUFFER
                 20   FORMAT (1H ,10A10)
                100   CONTINUE
                      STOP
                      END
```

```
6RM CHECK      0000000094          FILE INTERCHANGE
6RM CHECK      0000000095          FILE INTERCHANGE
6RM CHECK      0000000096          FILE INTERCHANGE
6RM CHECK      0000000097          FILE INTERCHANGE
6RM CHECK      0000000098          FILE INTERCHANGE
6RM CHECK      0000000099          FILE INTERCHANGE
6RM CHECK      0000000100          FILE INTERCHANGE
                        •
                        •
                        •
6RM CHECK      0000000001          FILE INTERCHANGE
6RM CHECK      0000000002          FILE INTERCHANGE
6RM CHECK      0000000003          FILE INTERCHANGE
6RM CHECK      0000000004          FILE INTERCHANGE
6RM CHECK      0000000005          FILE INTERCHANGE
6RM CHECK      0000000006          FILE INTERCHANGE
6RM CHECK      0000000007          FILE INTERCHANGE
6RM CHECK      0000000008          FILE INTERCHANGE
6RM CHECK      0000000009          FILE INTERCHANGE
6RM CHECK      0000000010          FILE INTERCHANGE
6RM CHECK      0000000011          FILE INTERCHANGE
6RM CHECK      0000000012          FILE INTERCHANGE
6RM CHECK      0000000013          FILE INTERCHANGE
6RM CHECK      0000000014          FILE INTERCHANGE
6RM CHECK      0000000015          FILE INTERCHANGE
6RM CHECK      0000000016          FILE INTERCHANGE
6RM CHECK      0000000017          FILE INTERCHANGE
6RM CHECK      0000000018          FILE INTERCHANGE
6RM CHECK      0000000019          FILE INTERCHANGE
```

```
       JOBOAGQ.   73/09/07.BAR ILAN UNIVERSITY.

11.27.23.JOB,CM55000.
11.27.23.ACCOUNT(YP)
11.27.23.COBOL(B=LGO1)
11.27.24.COMPILING SAMPLE
11.27.26. 000 E AND
11.27.26.    053000B SCM USED
11.27.26.       .251 CP SECONDS COMPILATION TIME
11.27.26.END COBOL
11.27.26.LGO1.
11.27.27.FILE(TEST,BT=C,RT=F,FL=100)
11.27.27.REWIND(TEST)
11.27.27.FTN.
11.27.28.       .052 CP SECONDS COMPILATION TIME
11.27.28.LDSET(FILES=TEST)
11.27.28.LGO(TEST)
11.27.29.    STOP
11.27.29.CP         1.308 SEC.
11.27.29.CM         0.006 KWH.
11.27.29.MS         0.487 KPR.
11.27.44.LP         0.233 KLN.
```

Figure 26-1. 6RM Examples (Continued)

CREATE File TAPE1

BT=C
RT=W

```
CHK

      PROGRAM CHK (TAPE1)
      DIMENSION IMAGE (100)
      DO 100 I=10,100,5
      DO 50 J=1,I
   50 IMAGE(J) = I
      WRITE (1) (IMAGE(K),K=1,I)    — Binary Write
  100 CONTINUE
      END
```

Retrieve Variable LTh. Records from TAPE1

```
      PROGRAM GET (TAPE1,OUTPUT)
      EXTERNAL EOI
      COMMON /LABEL/IMAGE(100),FIT(35),IEND
      CALL FILESQ (FIT,3LLFN,5LTAPE1)
      CALL OPENM (FIT,5LINPUT)
      IEND = 0
      DUM = 0
    5 IF (IEND .NE. 0) STOP
      LTH = 0
      CALL GET (FIT,IMAGE,DUM,DUM,DUM,LTH,EOI)
      LTH = IFETCH (FIT,2LRL)
      LLTH = LTH/10
      PRINT 10, LLTH,(IMAGE(I),I=1,LLTH)
   10 FORMAT (1H ,//,*LLTH =*,O10,/(5X,O20))
      GO TO 5
      END
```

```
      SUBROUTINE EOI
      COMMON /LABEL/ IMAGE(100),FIT(35),IEND
      IEND = 1
      CALL CLOSEM (FIT)
      RETURN
      END
```

```
LLTH =0000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012
     00000000000000000012

LLTH =0000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017
     00000000000000000017

     •
     •
     •
```

```
JC02AGI.    73/09/07.BAR ILAN UNIVERSITY.

11.16.43.JOB2.
11.16.44.ACCOUNT(YP)
11.16.44.GET(TAPE1)
11.16.44.FILE(TAPE1,RT=W,BT=C,MRL=1000)
11.16.44.FTN.
11.16.46.      .133 CP SECONDS COMPILATION TIME
11.16.46.LDSET(FILES=TAPE1)
11.16.46.LGO.
11.16.48.     STOP
11.16.48.CP        1.167 SEC.
11.16.48.CM        0.004 KWH.
11.16.48.MS        0.313 KPR.
11.17.40.LP        1.237 KLN.
```

Figure 26-1. 6RM Examples (Continued)

TAPE1

W TYPE Record Control Word

```
        - FILE DUMP -        TDUMP(I=TAPE1,O)                                        3/09/07. 11.13.32. PAGE     1.
F  1 R   1 W      0- 4000 0000 0000 0000 0012   0000 0000 0000 0000 0012   0000 0000 0000 0000 0012   0000 0000 0000 0000 0012
F  1 R   1 W      4- 0000 0000 0000 0000 0012   0000 0000 0060 0000 0012   0000 0000 0000 0000 0012   0000 0000 0006 0000 0012
F  1 R   1 W     10- 0000 0000 0000 0000 0012   0000 0000 0000 0000 0012   0000 0000 0000 0000 0012   0000 0000 0013 0000 0017
F  1 R   1 W     14- 0000 0000 0000 0000 0017   0000 0000 0000 0000 0017   0000 0000 0000 0000 0017   0000 0000 0000 0000 0017
-- ABOVE LINE REPEATED --
F  1 R   1 W     30- 0000 0000 0000 0000 0017   0000 0000 0000 0000 0017   0000 0000 0000 0000 0017   0000 0000 0020 0000 0024
F  1 R   1 W     34- 0000 0000 0000 0000 0024   0000 0000 0000 0000 0024   0000 0000 0000 0000 0024   0000 0000 0000 0000 0024
-- ABOVE LINE REPEATED --
F  1 R   1 W     60- 4000 0000 0025 0000 0031   0000 0000 0000 0000 0031   0000 0000 0000 0000 0031   0000 0000 0000 0000 0031
F  1 R   1 W     64- 0000 0000 0000 0000 0031   0000 0000 0000 0000 0031   0000 0000 0000 0000 0031   0000 0000 0000 0000 0031
-- ABOVE LINE REPEATED --
F  1 R   1 W    110- 0000 0000 0000 0000 0031   0000 0000 0000 0000 0031   0000 0000 0032 0000 0036   0000 0000 0000 0000 0036
F  1 R   1 W    114- 0000 0000 0000 0000 0036   0000 0000 0000 0000 0036   0000 0000 0000 0000 0036   0000 0000 0000 0000 0036
-- ABOVE LINE REPEATED --
F  1 R   1 W    150- 0000 0000 0000 0000 0036   4000 0000 0037 0000 0043   0000 0000 0000 0000 0043   0000 0000 0000 0000 0043
F  1 R   1 W    154- 0000 0000 0000 0000 0043   0000 0000 0000 0000 0043   0000 0000 0000 0000 0043   0000 0000 0000 0000 0043
-- ABOVE LINE REPEATED --
F  1 R   1 W    214- 0000 0000 0000 0000 0043   4000 0000 0044 0000 0050   0000 0000 0000 0000 0050   0000 0000 0000 0000 0050
F  1 R   1 W    220- 0000 0000 0000 0000 0050   0000 0000 0000 0000 0050   0000 0000 0000 0000 0050   0000 0000 0000 0000 0050
-- ABOVE LINE REPEATED --
F  1 R   1 W    264- 0000 0000 0000 0000 0050   0000 0000 0000 0000 0050   0000 0000 0051 0000 0055   0000 0000 0000 0000 0055
F  1 R   1 W    270- 0000 0000 0000 0000 0055   0000 0000 0000 0000 0055   0000 0000 0000 0000 0055   0000 0000 0000 0000 0055
-- ABOVE LINE REPEATED --
F  1 R   1 W    344- 0000 0000 0056 0000 0062   0000 0000 0000 0000 0062   0000 0000 0000 0000 0062   0000 0000 0000 0000 0062
F  1 R   1 W    350- 0000 0000 0000 0000 0062   0000 0000 0000 0000 0062   0000 0000 0000 0000 0062   0000 0000 0000 0000 0062
-- ABOVE LINE REPEATED --
F  1 R   1 W    424- 0000 0000 0000 0000 0062   0000 0000 0000 0000 0062   0000 0000 0000 0000 0062   0000 0000 0063 0000 0067
F  1 R   1 W    430- 0000 0000 0000 0000 0067   0000 0000 0000 0000 0067   0000 0000 0000 0000 0067   0000 0000 0000 0000 0067
-- ABOVE LINE REPEATED --
F  1 R   1 W    514- 0000 0000 0000 0000 0067   0000 0000 0000 0000 0067   0000 0000 0000 0000 0067   0000 0000 0070 0000 0074
F  1 R   1 W    520- 0000 0000 0000 0000 0074   0000 0000 0000 0000 0074   0000 0000 0000 0000 0074   0000 0000 0000 0000 0074
-- ABOVE LINE REPEATED --
F  1 R   1 W    614- 0000 0000 0075 0000 0101   0000 0000 0000 0000 0101   0000 0000 0000 0000 0101   0000 0000 0000 0000 0101
F  1 R   1 W    620- 0000 0000 0000 0000 0101   0000 0000 0000 0000 0101   0000 0000 0000 0000 0101   0000 0000 0000 0000 0101
-- ABOVE LINE REPEATED --
F  1 R   1 W    714- 0000 0000 0000 0000 0101   0000 0000 0000 0000 0101   0000 0000 0102 0000 0106   0000 0000 0000 0000 0106
F  1 R   1 W    720- 0000 0000 0000 0000 0106   0000 0000 0000 0000 0106   0000 0000 0000 0000 0106   0000 0000 0000 0000 0106
-- ABOVE LINE REPEATED --
F  1 R   1 W   1024- 0000 0000 0000 0000 0106   4000 0000 0107 0000 0113   0000 0000 0000 0000 0113   0000 0000 0000 0000 0113
F  1 R   1 W   1030- 0000 0000 0000 0000 0113   0000 0000 0000 0000 0113   0000 0000 0000 0000 0113   0000 0000 0000 0000 0113
-- ABOVE LINE REPEATED --
F  1 R   1 W   1140- 0000 0000 0000 0000 0113   0000 0000 0114 0000 0120   0000 0000 0000 0000 0120   0000 0000 0000 0000 0120
F  1 R   1 W   1144- 0000 0000 0000 0000 0120   0000 0000 0000 0000 0120   0000 0000 0000 0000 0120   0000 0000 0000 0000 0120
-- ABOVE LINE REPEATED --
F  1 R   1 W   1260- 0000 0000 0000 0000 0120   0000 0000 0000 0000 0120   0000 0000 0121 0000 0125   0000 0000 0000 0000 0125
F  1 R   1 W   1264- 0000 0000 0000 0000 0125   0000 0000 0000 0000 0125   0000 0000 0000 0000 0125   0000 0000 0000 0000 0125
-- ABOVE LINE REPEATED --
F  1 R   1 W   1410- 4000 0000 0126 0000 0132   0000 0000 0000 0000 0132   0000 0000 0000 0000 0132   0000 0000 0000 0000 0132
F  1 R   1 W   1414- 0000 0000 0000 0000 0132   0000 0000 0000 0000 0132   0000 0000 0000 0000 0132   0000 0000 0000 0000 0132
-- ABOVE LINE REPEATED --
F  1 R   1 W   1540- 0000 0000 0000 0000 0132   0000 0000 0000 0000 0132   0000 0000 0000 0000 0132   0000 0000 0133 0000 0137
F  1 R   1 W   1544- 0000 0000 0000 0000 0137   0000 0000 0000 0000 0137   0000 0000 0000 0000 0137   0000 0000 0000 0000 0137
-- ABOVE LINE REPEATED --
F  1 R   1 W   1700- 0000 0000 0000 0000 0137   0000 0000 0000 0000 0137   0000 0000 0000 0000 0137   0000 0000 0140 0000 0144
F  1 R   1 W   1704- 0000 0000 0000 0000 0144   0000 0000 0000 0000 0144   0000 0000 0000 0000 0144   0000 0000 0000 0000 0144
-- ABOVE LINE REPEATED --

F  1 R   1 W   2050- 1000 0000 0145 0000 0000
```

Figure 26-1. 6RM Examples (Continued)

```
      PROGRAM RANDOM (INPUT,TAPE1=INPUT,TAPE3,OUTPUT)
      DIMENSION INDEX(11), IMAGE(8)
      N = 1
      CALL OPENMS (3,INDEX,11,0)
      DO 100 N = 1,10
      READ (1,10) (IMAGE(I), I = 1,8)
   10 FORMAT (I10,7A10)
      IREC = IMAGE (1)
      CALL WRITMS (3,IMAGE,8,IREC)
  100 CONTINUE
      END
```

```
10
5   This is Record Number FIVE   **---**
2                         TWO
7                         SEVEN
6          .              SIX       .
10         .              TEN       .
1          .              ONE       .
3          .              THREE
8                         EIGHT
4                         FOUR
9                         NINE
```

READMS / WRITMS

USING FTN 4.0 & RUN 2.3

```
      PROGRAM RET (TAPE3,INPUT,OUTPUT)
      DIMENSION INDEX(11),IMAGE(8)
      CALL OPENMS(3,INDEX,11,0)
    5 READ 10,I
   10 FORMAT (I10)
      IF (I .EQ. 0 .OR. I .GT. 10) GO TO 100
      CALL READMS (3,IMAGE,8,I)
      PRINT 20, (IMAGE(J), J=1,5)
   20 FORMAT (I10,7A10)
      GO TO 5
  100 STOP
      END
```

```
10
5
1
7
2
10
8
0
```

```
         5      THIS IS RECORD NUMBER FIVE             ...................
         1      THIS IS RECORD NUMBER ONE              ...................
         7      THIS IS RECORD NUMBER SEVEN            ...................
         2      THIS IS RECORD NUMBER TWO              ...................
        10      THIS IS RECORD NUMBER TEN              ...................
         8      THIS IS RECORD NUMBER EIGHT            ...................
```

Figure 26-1. 6RM Examples (Continued)

```
- L DUMP -            (DUMP(I=TAPE3,O)                    F11X                 1970-07-0   11.19.31   PAGE  1
F  1 R   1 W    0- 0000 0000 0001 0000 0001   7000 0001 3000 0000 0233   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F  1 R   1 W    4- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
-- ABOVE LINE REPEATED --
F  1 R   1 W  100- 0000 0000 0000 0000 0010   0000 0000 0000 0000 0005   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503
F  1 R   1 W  104- 1722 0455 1625 1502 0522   5506 1126 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747
F  1 R   1 W  110- 4747 4747 4747 4747 4747   0000 0000 0000 0000 0010   0000 0000 0000 0000 0002   5555 5555 5555 5555 5524
F  1 R   1 W  114- 1011 2355 1123 5522 0503   1722 0455 1625 1502 0522   5524 2717 5555 5555 5555   5555 5555 5555 5555 5547
F  1 R   1 W  120- 4747 4747 4747 4747 4747   4747 4747 4747 4747 4747   0000 0000 0000 0000 0010   0000 0000 0000 0000 0007
F  1 R   1 W  124- 5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522   5523 0526 0516 5555 5555
F  1 R   1 W  130- 5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747   0000 0000 0000 0000 0010
F  1 R   1 W  134- 0000 0000 0000 0000 0006   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   1 W  140- 5523 1130 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
F  1 R   1 W  144- 0000 0000 0000 0000 0010   0000 0000 0000 0000 0012   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503
F  1 R   1 W  150- 1722 0455 1625 1502 0522   5524 0516 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747
F  1 R   1 W  154- 4747 4747 4747 4747 4747   0000 0000 0000 0000 0010   0000 0000 0000 0000 0001   5555 5555 5555 5555 5524
F  1 R   1 W  160- 1011 2355 1123 5522 0503   1722 0455 1625 1502 0522   5517 1605 5555 5555 5555   5555 5555 5555 5555 5547
F  1 R   1 W  164- 4747 4747 4747 4747 4747   4747 4747 4747 4747 4747   0000 0000 0000 0000 0010   0000 0000 0000 0000 0003
F  1 R   1 W  170- 5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522   5524 1022 0515 5555 5555
F  1 R   1 W  174- 5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747   0000 0000 0000 0000 0010
F  1 R   1 W  200- 0000 0000 0000 0000 0010   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   1 W  204- 5505 1107 1024 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
F  1 R   1 W  210- 0000 0000 0000 0000 0010   0000 0000 0000 0000 0004   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503
F  1 R   1 W  214- 1722 0455 1625 1502 0522   5506 1725 2255 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747
F  1 R   1 W  220- 4747 4747 4747 4747 4747   0000 0000 0000 0000 0010   0000 0000 0000 0000 0011   5555 5555 5555 5555 5524
F  1 R   1 W  224- 1011 2355 1123 5522 0503   1722 0455 1625 1502 0522   5516 1116 0555 5555 5555   5555 5555 5555 5555 5547
F  1 R   1 W  230- 4747 4747 4747 4747 4747   4747 4747 4747 4747 4747   0000 0000 0000 0000 0013   0000 0000 0000 0000 0233
F  1 R   1 W  234- 2000 0001 0000 0000 0156   2000 0001 0000 0000 0112   2000 0001 0000 0000 0167   2000 0001 0000 0000 0211
F  1 R   1 W  240- 2000 0001 0000 0000 0131   2000 0001 0000 0000 0134   2000 0001 0000 0000 0123   2000 0001 0000 0000 0233
F  1 R   1 W  244- 2000 0001 0000 0000 0222   2000 0001 0000 0000 0145   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
F  1 R   1 W  250- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
-- ABOVE LINE REPEATED --

F  1 R   1 W 1774- 0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000   0000 0000 0000 0000 0000
-- END OF INFORMATION --

-- END OF DUMP --
```

FTN 4.0 WRITMS Generated File = WORD ADDRESSABLE File

RT=W

Note Each user's logical record is Blocked into 1 System logical record.

Figure 26-1. 6RM Examples (Continued)

```
        - FILE DUMP -          DUMP(L=TAPE3,0)

F  1 R   1 W     0- 0000 0000 0000 0000 0005   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   1 W     4- 5506 1126 0555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   2 W     0- 0000 0000 0000 0000 0002   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   2 W     4- 5524 2717 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   3 W     0- 0000 0000 0000 0000 0007   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   3 W     4- 5523 0526 0516 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   4 W     0- 0000 0000 0000 0000 0006   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   4 W     4- 5523 1130 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   5 W     0- 0000 0000 0000 0000 0012   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   5 W     4- 5524 0516 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   6 W     0- 0000 0000 0000 0000 0001   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   6 W     4- 5517 1605 5555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R   7 W     0- 0000 0000 0000 0000 0003   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R   7 W     4- 5524 1022 0505 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R  10 W     0- 0000 0000 0000 0000 0010   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R  10 W     4- 5505 1107 1024 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R  11 W     0- 0000 0000 0000 0000 0004   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R  11 W     4- 5506 1725 2255 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R  12 W     0- 0000 0000 0000 0000 0011   5555 5555 5555 5555 5524   1011 2355 1123 5522 0503   1722 0455 1625 1502 0522
F  1 R  12 W     4- 5516 1116 0555 5555 5555   5555 5555 5555 5555 5547   4747 4747 4747 4747 4747   4747 4747 4747 4747 4747
-- END OF RECORD --

F  1 R  13 W     0- 0000 0000 0000 0000 0001   0000 0000 0000 0000 0006   0000 0000 0000 0000 0002   0000 0000 0000 0000 0007
F  1 R  13 W     4- 0000 0000 0000 0000 0011   0000 0000 0000 0000 0001   0000 0000 0000 0000 0004   0000 0000 0000 0000 0003
F  1 R  13 W    10- 0000 0000 0000 0000 0010   0000 0000 0000 0000 0012   0000 0000 0000 0000 0005
-- END OF RECORD --

-- END OF INFORMATION --

-- END OF DUMP --
```

RUN 2.3 WRITMS Generated File

Note Each user's logical record = System logical record

Figure 26-1. 6RM Examples (Continued)

SORTMRG 4.0

```
'T  .J'J'' 4.0

SC  T.
FI  ,I IPUT=INPUT(CR),OUTPUT=DISK1(CR)
3   ELD,K1(1,3,DISPLAY),K2(16,1,DISPLAY)
  KY,K1(A,COBOL6),K2(A,COBOL6)
>  END.
```

```
SORTAIO.   73/09/07.BAR ILAN UNIVERSITY.

13.10.16.SORTEX,CM50000.
13.10.15.ACCOUNT,YP).
13.10.16.FILE(INPUT,RT=C,RT=Z,MBL=80,MRL=90)
13.10.17.FILE(DISK1,AT=C,RT=7,MBL=640,MRL=80)
13.10.17.SORTMRG.
13.10.19. ** INSERTIONS DURING INPUT      **********0
13.10.19. ** DELETIONS DURING INPUT       **********0
13.10.19. ** TOTAL RECORDS SORTED         *********32
13.10.19. ** INSERTIONS DURING OUTPUT     **********0
13.10.19. ** DELETIONS DURING OUTPUT      **********0
13.10.19. ** TOTAL RECORDS OUTPUT         *********32
13.10.19. **END SORT RUN
13.10.19.REWIND(DISK1)
13.10.19.COPYSPF(DISK1,OUTPUT)
13.10.18.COPY COMPLETE.
13.10.19.CP         0.092 SEC.
13.10.19.MS         0.026 KPR.
13.10.25.LP         0.092 KLN.
```

```
001234
001122334455667788996633225589774411
100000010000001
100000010000002
100000010000003
100000010000005
100000010000009
12304654064657987360164651651654012356897
1477412255586302101654987324024357487353
20202002545024026578102762162453887652313289966237
222001654987365021321654986463316798 7
400000000000001
400000000000002
400000000000003
400000000000004
430000000000005
400000000000006
43210565214632701793465343434376738387964313135673 87
46132404987633120751498435132102132146 5577
4654201479686321564013579497654321621321 92
65002147896313242244224242424353355353533557
700000000000001
700000000000002
700000000000003
700000000000004
700000000000005
700000000000007
700000000000008
700000000000009
765343124324679976543213210623265497987433210013554 
798965326532132401330 04426522
9879654631 0987654321365797986543 1+316579876543213 13
```

Figure 26-1. 6RM Examples (Continued)

```
                    60 0016                    COBOL     V402118     06/01/72 P01.18.00.        PAGE     1

              IDENTIFICATION DIVISION.
                  PROGRAM-ID. SISSDA.
              ENVIRONMENT DIVISION.
              CONFIGURATION SECTION.
              SOURCE-COMPUTER. 6400.
              OBJECT-COMPUTER. 6400.
              INPUT-OUTPUT SECTION.
              FILE-CONTROL.
                  SELECT FILE1 ASSIGN TO TAPE1
                  ORGANIZATION IS INDEXED SEQUENTIAL
                  FILE-LIMIT IS 500
                  SYMBOLIC KEY IS THE-KEY.
              DATA DIVISION.
                  FILE SECTION.
              FD  FILE1  LABEL RECORDS ARE OMITTED
                    DATA RECORD IS REC.
              01  REC SIZE IS 100 CHARACTERS.
                  02  IMAGE.
                    03  IM1 SIZE IS 6.
                    03  THE-KEY PIC 9(4).
                    03  FILLER SIZE IS 10.
                    03  IM2 SIZE IS 80.
              WORKING-STORAGE SECTION.
              77  ACT-KEY PIC 9(4).
              77  NO-RECS PIC 9(4) VALUE IS 500.

              PROCEDURE DIVISION.
              CREATE.
                  DISPLAY ≠START CREATING FILE≠.
                  OPEN OUTPUT FILE1.
                  MOVE SPACES TO IMAGE.
                  MOVE ≠RECORD≠ TO IM1.
                  MOVE ≠6RM FILE TEST≠ TO IM2.
                  MOVE 0 TO ACT-KEY.
                  PERFORM WR-REC NO-RECS TIMES.
                  CLOSE FILE1.
                  DISPLAY ≠FINISHED CREATING FILE≠.
                  STOP RUN.
              WR-REC.
                  ADD 1 TO ACT-KEY.
                  MOVE ACT-KEY TO THE-KEY.
                  IF ACT-KEY EQ 1 DISPLAY REC.
                  IF ACT-KEY EQ NO-RECS DISPLAY REC.
                  WRITE REC INVALID KEY DISPLAY ≠THE KEY = ≠ THE-KEY STOP RUN.
 SISSDA  LENGTH IS  000163
FIELD LENGTH NEEDED FOR COBOL 052700


 START CREATING FILE
 RECORD0001          6RM FILE TEST
 RECORD0500          6RM FILE TEST
 FINISHED CREATING FILE
```

CREATE SIS FILE

```
 06/01/72  SCOPE 3.4   SVSNIC2LEVEL CS  04/27/72
01.17.47.JOB002C
01.17.47.JOB,T200,CM60000.   CS
01.17.47.FILE(TAPE1,NL=2,TBL=630,MBL=1000)
01.17.51.      1.791 RT SECONDS LOAD TIME
01.17.55.COBOL.
01.17.59.      2.710 RT SECONDS LOAD TIME
01.18.02.COMPILING SISSDA
01.18.13. 000 E AND     T/U  DIAGNOSTICS ISSUED
01.18.13. FIELD LENGTH NEEDED FOR COBOL 052700
01.18.13.      .645 CP SECONDS COMPILATION TIME
01.18.13.END COBOL
01.18.13.LGO.
01.18.24.      10.109 RT SECONDS LOAD TIME
01.18.44.CATALOG(TAPE1,SIS,ID=DT,FO=1S)
01.18.45.INITIAL CATALOG
01.18.47.PF CYCLE NO. = 001
01.18.51.RP = 010 DAYS
01.18.52.CP         4.616 SEC.
01.18.52.PP        24.539 SEC.
01.18.52.IO         1.407 SEC.
```

Figure 26-1. 6RM Examples (Continued)

```
AD 0016                                   COBOL    V402118    06/01/72  01.29.20.     PAGE    1
```

SIS SEQUENTIAL ACCESS

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SISSDA
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. 6400.
OBJECT-COMPUTER. 6400.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT FILE1 ASSIGN TO TAPE1
    ORGANIZATION IS INDEXED SEQUENTIAL
    FILE-LIMIT IS 1000
    ACCESS MODE IS SEQUENTIAL
    SYMBOLIC KEY IS THE-KEY.
DATA DIVISION.
FILE SECTION.
FD  FILE1 LABEL RECORDS ARE OMITTED
    DATA RECORD IS REC.
01  REC PIC X(100).
WORKING-STORAGE SECTION.
77  ACT-KEY PIC 9(4) VALUE IS 0.
77  NO-RECS PIC 9(4) VALUE IS 1000.
77  THE-KEY PIC 9(4).


PROCEDURE DIVISION.
STARTIT.
    OPEN INPUT FILE1.
    DISPLAY "ACCESS FILE SEQUENTIALLY".
AGAIN.
    READ FILE1 AT END DISPLAY "THE END" CLOSE FILE1 STOP RUN.
    DISPLAY REC.
    GO TO AGAIN.
SISSDA  LENGTH IS  000063
FIELD LENGTH NEEDED FOR COBOL 052700
```

```
ACCESS FILE SEQUENTIALLY
RECORD0001          6RM FILE TEST
RECORD0002          6RM FILE TEST
RECORD0003          6RM FILE TEST
RECORD0004          6RM FILE TEST
RECORD0005          6RM FILE TEST
RECORD0006          6RM FILE TEST
RECORD0007          6RM FILE TEST
RECORD0008          6RM FILE TEST
RECORD0009          6RM FILE TEST
RECORD0010          6RM FILE TEST
RECORD0011          6RM FILE TEST
RECORD0012          6RM FILE TEST
RECORD0013          6RM FILE TEST
RECORD0014          6RM FILE TEST
RECORD0015          6RM FILE TEST
RECORD0016          6RM FILE TEST
RECORD0017          6RM FILE TEST
RECORD0018          6RM FILE TEST
RECORD0019          6RM FILE TEST
RECORD0020          6RM FILE TEST
RECORD0021          6RM FILE TEST
RECORD0022          6RM FILE TEST
        .
        .
        .
RECORD0499          6RM FILE TEST
RECORD0500          6RM FILE TEST
THE END
```

```
 06/01/72  SCOPE 3.4   SVSN102LEVEL CS  04/27/72
01.28.14.JOB002F
01.28.14.JOB,T200,CM60000.
01.28.15.ATTACH(TAPE1,SIS,ID=DT,FO=IS)
01.28.16.PF CYCLE NO. = 001
01.28.16.COBOL.
01.28.19.      2.119 PT SECONDS LOAD TIME
01.28.21.COMPILING SISSDA
01.28.31. 000 E AND     T/U  DIAGNOSTICS ISSUED
01.28.31. FIELD LENGTH NEEDED FOR COBOL 052700
01.28.31.        .461 CP SECONDS COMPILATION TIME
01.28.31.END COBOL
01.28.31.LGO.
01.28.43.    11.090 PT SECONDS LOAD TIME
01.29.15.CP       4.549 SEC.
01.29.15.PP      14.741 SEC.
01.29.15.IO       1.352 SEC.
```

Figure 26-1. 6RM Examples (Continued)

```
AO 0016                                        COBOL    V402118    06/01/72 *01.20.10.    PAGE    1

        IDENTIFICATION DIVISION.
        PROGRAM-ID. SISSOA
        ENVIRONMENT DIVISION.
        CONFIGURATION SECTION.
        SOURCE-COMPUTER. 6400.
        OBJECT-COMPUTER. 6400.
        INPUT-OUTPUT SECTION.
        FILE-CONTROL.
          SELECT FILE1 ASSIGN TO TAPE1
            ORGANIZATION IS INDEXED SEQUENTIAL
            FILE-LIMIT IS 1000
            ACCESS MODE IS RANDOM
            SYMBOLIC KEY IS THE-KEY.
        DATA DIVISION.
        FILE SECTION.
        FD  FILE1 LABEL RECORDS ARE OMITTED
            DATA RECORD IS REC.
        01  REC PIC X(100).
        WORKING-STORAGE SECTION.
        77  ACT-KEY PIC 9(4) VALUE IS 0.
        77  NO-RECS PIC 9(4) VALUE IS 1000.
        77  THE-KEY PIC 9(4).


        PROCEDURE DIVISION.
        STARTIT.
            OPEN INPUT FILE1.
            DISPLAY ≠READ EVERY 100TH RECORD≠.
        AGAIN.
            ADD 100 TO ACT-KEY.
            MOVE ACT-KEY TO THE-KEY.
            IF  ACT-KEY GREATER THAN NO-RECS GO TO RESUME.
            READ FILE1 INVALID KEY DISPLAY ≠THE BAD KEY = ≠ ACT-KEY
            STOP RUN.
            DISPLAY REC.
            GO TO AGAIN.
        RESUME.
            CLOSE FILE1.
            DISPLAY ≠FINISHED≠
            STOP RUN.
  SISSOA  LENGTH IS  000116
 IELD LENGTH NEEDED FOR COBOL 052700
```

SIS RANDOM ACCESS

```
 06/01/72  SCOPE 3.4    SUSNI02LEVEL CS  04/27/72
01.20.05.JOB002F
01.20.06.JOB,T200,CM60000.
01.20.06.ATTACH(TAPE1,SIS,ID=DT,FO=IS)
01.20.10.PF CYCLE NO. = 001
01.20.10.COBOL.
01.20.17.     4.943 PT SECONDS LOAD TIME
01.20.21.COMPILING SISSOA
01.20.32. 000 E AND     I/U  DIAGNOSTICS ISSUED
01.20.32. FIELD LENGTH NEEDED FOR COBOL 052700
01.20.33.      .609 CP SECONDS COMPILATION TIME
01.20.33.END COBOL
01.20.33.LGO.
01.20.45.     11.359 RT SECONDS LOAD TIME
01.20.56.CP       3.830 SEC.
01.20.56.PP      18.709 SEC.
01.20.56.IO       1.247 SEC.
```

```
READ EVERY 100TH RECORD
RECORD0100          6RM FILE TEST
RECORD0200          6RM FILE TEST
RECORD0300          6RM FILE TEST
RECORD0400          6RM FILE TEST
RECORD0500          6RM FILE TEST
THE BAD KEY = 0600
```

Figure 26-1. 6RM Examples (Continued)

```
AO 0016                                  COBOL    V402118    06/01/72 *01.53.42.      PAGE

      IDENTIFICATION DIVISION.
      PROGRAM-ID. SISSDA.
      ENVIRONMENT DIVISION.
      CONFIGURATION SECTION.
      SOURCE-COMPUTER. 6400.
      OBJECT-COMPUTER. 6400.
      INPUT-OUTPUT SECTION.
      FILE-CONTROL.
        SELECT FILE1 ASSIGN TO TAPE1
          ORGANIZATION IS INDEXED SEQUENTIAL
          FILE-LIMIT IS 1000
          ACCESS MODE IS RANDOM
          SYMBOLIC KEY IS THE-KEY.
      DATA DIVISION.
      FILE SECTION.
      FD  FILE1 LABEL RECORDS ARE OMITTED
          DATA RECORD IS REC.
      01  REC PIC X(100).
      WORKING-STORAGE SECTION.
      77  ACT-KEY PIC 9(4) VALUE IS 0.
      77  NO-RECS PIC 9(4) VALUE IS 1000.
      01  AAAAA
ERROR       02      FILLER PIC X(6) VALUE IS ≠RECORD≠.
            02  THE-KEY PIC 9(4).
            02  FILLER PIC X(10) VALUE IS SPACES.
            02  DUM PIC X(80) VALUE IS ≠REWRITTEN RECORD≠.

      PROCEDURE DIVISION.
      STARTIT.
          OPEN I-O FILE1.
          DISPLAY  ≠REWRITE EVERY 100TH RECORD≠.
      AGAIN.
          ADD 100 TO ACT-KEY.
          MOVE ACT-KEY TO THE-KEY.
          IF ACT-KEY GREATER THAN NO-RECS GO TO RESUME.
          REWRITE REC FROM AAAAA INVALID KEY
                  DISPLAY ≠THE BAD KEY =≠ ACT-KEY STOP RUN.
          DISPLAY REC.
          GO TO AGAIN.
      RESUME.
          CLOSE FILE1.
          DISPLAY ≠FINISHED≠
          STOP RUN.
 SISSDA  LENGTH IS  000135
FIELD LENGTH NEEDED FOR COBOL 052700
```

SIS UPDATE

```
REWRITE EVERY 100TH RECORD
RECORD0100         REWRITTEN RECORD
RECORD0200         REWRITTEN RECORD
RECORD0300         REWRITTEN RECORD
RECORD0400         REWRITTEN RECORD
RECORD0500         REWRITTEN RECORD
THE BAD KEY =0600
```

```
   06/01/72  SCOPE 3.4   SVSN102LEVEL CS  04/27/72
01.53.30..IOB002M
01.53.30. IOB.T200.CM60000.
01.53.31.ATTACH(TAPE1.SIS.IO=OT.FO=IS)
01.53.34.PF CYCLE NO. = 001
01.53.34.COBOL.
01.53.40.      6.187 RT SECONDS LOAD TIME
01.53.45.COMPILING SISSDA
01.54.05. 001 E AND     T/U  DIAGNOSTICS ISSUED
01.54.05. FIELD LENGTH NEEDED FOR COBOL 052700
01.54.09.       .703 CP SECONDS COMPILATION TIME
01.54.09.END COBOL
01.54.09.LGO.
01.54.29.     19.234 RT SECONDS LOAD TIME
01.54.41.CP      4.190 SEC.
01.54.41.PP     23.726 SEC.
01.54.41.IO      1.391 SEC.
```

Figure 26-1. 6RM Examples (Continued)

block type — RT = Record type

BT = C binary old 6000 type

K } E } MT's Xrecs/block

I old FORTRAN type

See examples in Figure 26-1.

26.2 PROCESSOR OVERLAYS

The other processors mentioned previously (namely, STS, EMG, RPV, PFE, ACE, PRM, CKP, REQ, and DMP) are all overlays in the PP routine SFP except CKP, REQ and DMP. See Section 4, PP Resident, for a SFP description. See Section 5 for a discussion of DMP, and Section 22 for a discussion of CKP. The other routines are described subsequently.

26.2.1 Status Processor - STS

RA+1 =

18	6	12	6	18
STS	0	FUNCTION CODE	0	ADR

Function 01 - Return mass storage devices status Returns status of mass storage devices starting at address+1 of address contained in bits 0-17 of program call. Return is defined by address:

ADR =

	12	12	23	1
0	LL	LR	0	A

where:

LL = Number of words, excluding this header word, to be used for return information; must be set by user to other than 0.

LR = Number of status words returned.

A = Auto recall reply: set to 0 by user and set to 1 when request is complete

The mass storage device status is returned, 1 word per device, in the following format:

ADR+N=

3	9	12	12	6	6	12
0	STATUS	DEV TYPE	EST ORD	CHAN	EQ	PRUS

where:

STATUS = 000 - Not available, off, not in use

040 - Unloaded pack

120 - KRONOS system routines

140 - KRONOS system routines on pack

620 - Contains permanent files

640 - Pack with permanent files

700 - KRONOS system and permanent files

740 - KRONOS system and perm files on pack

DEV TYPE = SCOPE 3.4 hardware mnemonic in display code.

AA - 6603 Disk System

AB - 6638 Disk System

AD - 865 Drum System

AF - 814 Disk System

AL - 821 Disk System

AM - 841 Disk System

AP - 854 Disk System

AY - 844 Disk System

PRUs = Number of PRUs/100 octal of space remaining on the device. A value of 7777 indicates at least 262,100 PRUs available.

Function 02 - Return file status.

Returns to the calling program the FNT/FST entries of files requested whose names are set in every third location starting with PARM+1 of address contained in the "PARM" field of the calling program. If the file exists, the file name will be replaced by the FNT/FST of KRONOS mapped into the SCOPE 3.4 FNT/FST. If the file does not exist, the file name will be zeroed out.

Format of location pointed to by "PARM":

PARM =

12	12	12	23	1
0	LL	LR	0	A

where:

LL = Number of words, excluding this header word, to be used for return information: must be set by user to a multiple of three

LR = Length of status area returned.

A = Auto recall reply: set to zero by user and set to one when request is complete.

Format of mapped 3 word KRONOS FNT/FST.

42	1	5	12
file name	0	CP	0

Devt	0	1st track	curtrack	0	cur sec

0	disp. code	Pem	0	CS
24	12	4	8	12

Function 03 -Return PRU count of file(s)

Returns to the calling program the number of PRUs of the files requested whose names are set in every second word starting at "PARM+1" of address contained in the "PARM" field of the calling program. If the file exists, the PRU count will be returned in bits 0-23 of the second word. If the file does not exist, the second word will be zero.

Format of location pointed to by "PARM".

T PARM =

12	12	12	23	1
0	LL	LR	0	A

where:

LL = Number of words, excluding this header, to be used: must be set by user to a multiple of 2.

LR = Length of status reply area.

A = Auto recall reply: set to zero by user and set to one when request is complete.

26.2.2 SDA/SIS Message Generator - EMG

Returns messages to SDA/SIS as requested by a message code contained in the PP call parameter area. EMG performs the function of the SCOPE 3.4 PP program "MSD".

ENTRY (IR - IR+4 = call to "MSD" with the format:

IR =

MSD	CP/AO	Message code	Return address

where:

MESSAGE CODE = Message ordinal of message to be returned

RETURN ADDRESS= CM address to return message beginning at (return address +1).

EXIT (Return Address) ≠ 0. Upon completion of message transfer, (return address) is set to:

IR =	Mess Code	Message -1	mess size	0	1

where:

MESS CODE = Message code issued in "MSD" call.

MESS SIZE = Message size in CM words of message returned.

26.2.3 Extract Error Text-(Used By COBOL) D00

D00 is a routine that will extract messages from specially created system text decks to aid in analyzing error situations resulting from a product set. By using an error number and the proper system text deck, an error diagnostic will be transmitted to the dayfile and/or to a specified CM buffer. All system text decks to be used must be MS resident.

	18	6	18	18
IR =	D00	CP NO	0	ADDRESS

	12	12	12	18	6
Address =	A	MSG NO	BUF SIZE	BUF ADDRESS	0

42	18
TEXT DECK NAME	INSERT CHARACTER

where:

A = 4000B - If insertions to messages.

A = 2000B - If dayfile message transfer.

A = 1000B - If CM buffer message transfer.

	36	12	12
IR =	ADDRESS	STATUS	1

where:

STATUS = 0 If transmittal to dayfile only.

= 7777B if error.

= CM words written if CM buffer transmittal.

26.2.4 Reprieve CP Program - RPV

RPV provides the ability for a CPU routine to get control back after a specified normal or abnormal termination condition. There are two cases in which RPV may be called, one is to initialize, the other is to reset.

The program recovery capability of SCOPE 3.4 is under KRONOS 2.1 to provide support of the SCOPE 3.4 products under KRONOS 2.1 through the use of the RECOVR macro contained in the SCOPE ACTCOM carried under KRONOS 2.1.

26.2.4.1 RECOVR Function

The RECOVR macro allows a user program to gain control at the time that normal or abnormal job termination procedures would otherwise occur. Initialization of RECOVR at the beginning of a program establishes the conditions under which control is to be regained and specifies the address of user recovery code. If the stated condition occurs during program execution, control returns to the user code. RECOVR macro expansion calls the SETUP. subroutine.

RECOVR is concerned with conditions that affect job execution. The conditions under which KRONOS will return control to the user, and the octal values that will select them in the call to RECOVR, are:

Arithmetic mode error	001
PP call error	002
Time limit	004
Operator drop	010
System abort	020
CP abort	040
Normal termination	100

Conditions can be combined as desired, with octal values up to 177 allowed in the flag field of the call to RECOVR.

At least five seconds of central processor time always will be available for user code execution. RECOVR makes the exchange jump package and RA + 1 contents available to the program if user recovery code is executed, and gives the user the option of having normal or abnormal job termination output.

Initialization of RECOVR within code at the beginning of a program results in an entry in a stack of requests for PP program RPV. Only one set of recovery conditions can exist within RPV, but RECOVR allows up to five user and system sets of flags and code for each program. The last RECOVR initialization will receive control first.

A checksum of the user recovery code can be requested during initialization. If flagged conditions subsequently occur, RECOVR will again checksum the code before returning control to it. This gives some assurance of user code integrity before it is executed.

RECOVR is initialized from a COMPASS program with:

```
RECOVR   name, flags,checksum
```

name	Address of code to be executed if flagged conditions occur; a return jump will be made to this location
flags	Octal value for conditions under which recovery code is to be executed, as outlined above; default is 77
checksum	Last word address of recovery code to be checksummed; 0 if no checksum

If one of the flagged conditions occurs, the address of the exchange jump package will be in register B1 and the RA address in B3. Register A1 will contain the address of the list of the parameters passed in B1-B3. Register B2 will contain a B; if the recovery code sets B2 to a non-zero value, or if the code contains an ENDRUN macro or an RA + 1 request for END, normal job termination procedures will follow. Otherwise, abnormal job termination procedures will follow recovery code execution.

If a program calling RECOVR contains overlays, both the call to RECOVR and the user recovery code should be a part of the level 0, 0 code.

The exchange jump package returned by RECOVR is in the format returned by DMP, with the system error code that caused recovery code execution in bits 0-17 of the first word. If the P register shows zero in the package because a mode error occurred, bits 31-47 of RA + 0 will contain the P register value. System error codes that may be returned are:

Normal termination	0
Requested time limit exceeded	1
Arithmetic mode error	2
PPU abort	3
CPU abort	4
PP call error	5
Operator dropped job	6
Program stop	7
File limit	10
Track limit	11
Mag tape limit	12
System abort	13

Both the FORTRAN and FORTRAN Extended languages contain RECOVR subroutines as detailed in their respective manuals.

26.2.4.2 RPV Function

RPV is the PPU portion of the RECOVR CP/PP package and is contained as a function in the PP program SFP. RPV should never be called directly, but utilize the CP portion for all program recovery work.

	18	6	4	7	6	18
RA+1 =	RPV	0	0	OPT	0	ADDRESS

where:

R = If set indicates a reset requested. Reset will be performed from the address last set.

OPT = Options when reprieve will be invoked. Each bit represents an error condition.

1 – Mode error
2 – RA + 1 error
4 – Time or storage limit exceeded
10 – Operator drop
20 – System abort
40 – CPU abort
100 – Normal termination

EXIT = If initialization option byte will be set in each, and exit address field will contain called address with bit 17 set.

26.2.4.3 RECOVR Error Messages

RECOVER – TOO MANY RECOVERY REQUESTS.– More than 5 recovery initializations occurred without a recovery being processed.

RECOVR – BAD CHECKSUM. – The post-recovery checksum of users recovery routine does not equal the pre-recovery checksum.

RECOVR – BAD ARGUMENT LIST. – Illegal parameters in pre-recovery initialization call.

26.2.4 RPV Error Messages

SFP/RPV INITIALIZATION ERROR. – If entry to RPV initialization without the Error Exit Return Address set.

REPRIEVE ABORTED BAD CHECKSUM. – Post-recovery checksum of RECOVR routine does not match pre-recovery checksum.

REPRIEVE ABORTED–VALIDATION TL. – Current time limit +5 seconds exceeds time that user has been validated for.

JOB REPRIEVED. * – The job has been successfully reprieved.

REPRIEVE ABORTED SYSTEM ERROR. – Error condition unknown to RPV.

REPRIEVE ROUTINE NOT IN FL. – The RECOVR CP routine not in users field length.

SFP/RPV CANNOT RESTORE PREVIOUS ERROR. – RPV cannot restore the error that caused the initial termination.

(PREVIOUS ERROR CONDITION RESET.) – RPV has restored the error condition that caused the termination.

26.2.5 Extend/Alter File Function - PFE

Alters the requested file to have an EOI recorded at the current position of the mass storage file.

	18	6	18	18
RA+1 =	PFE	P	0	PARM

where:

P = Set for Auto-Recall

Word contained at the address in the function call is:

	42	9	3	4	1	1
PARM =	LOGICAL FILE NAME	RET	0	F	O	C

where:

RET = If the "RC" and "RT" parameter defined in "0", a return code will be available to the user. The following codes will be returned:

000 - Function successful

003 - Unknown LFN

025 - File unavailable

0 = Options available are the following:
Bit 6 - Return code to user in RET

F = Function code for alter. Bits 2 - 5 = 0111.

C = Completion bit. Set when function is complete.

The PFE error message is: SFP/PFE ILLEGAL ALTER FUNCTION.

26.2.6 Advance Control Card - ACE

Reads/backspaces next/previous control card into RA+70B - RA+77B with the option to place the control card in the dayfile and/or to crack and store the control card parameters in SCOPE 3.4/KRONOS 2.1 format into RA+2 - RA+53B. If a read function is issued and the pointer is at the end of the control card record, and EOR status (bit 4 set in the function code) and RA+70B - RA+77B cleared. If a backspace function is issued and the pointer is at the beginning of the control card record, the pointer is not changed and an EOR status is returned. When function is complete, the completion bit (bit 0) is set and returned to the user.

	24	9	3	6	18
RA+1 =	ACEP	O	F	O	FUNC ADDRESS

where:

F = X01 - Crack parameters in KRONOS 2.1 format
= X10 - Crack parameters in SCOPE 3.4 format
= 1XX - Issue control card to dayfile

FUNC ADDRESS = CM word containing function to be performed.

	48	12
FUNC =	0	FUNCTION

where:

FUNC = 0010 - Read next control card and advance control card pointer.
FUNC = 0040 - Backspace to previous control card.

EXIT Completion bit set in FUNCTION. The following error message can be displayed:
SFP/ACE FUNCTION CODE UNDEFINED - The function at the FUNC ADDRESS is undefined.

26.2 7 SCOPE 3.4 Permission Checking Function - PRM

PRM will scan for an FNT entry whose address is contained in the call and, if found, will map the KRONOS 2.1 file permission bits into the SCOPE 3.4 permission bits and return to the user as a status. If the call address is out of range or the requested file does not exist, no diagnostic will be issued and no status will be returned to the user.

	24	18	18
RA+1	PRM	0	PARM ADDRESS

where:

PARM ADDRESS = CM address which contains the file name to search for

	42	9	8	1
PARM =	LFN	CODE	O	C

where:

CODE = A 5-bit code returned by PRM in bits 9-13. The rightmost 4 bits are the permission bits. The octal codes are:

01 - Read
02 - Extend
04 - Modify
10 - Control

The leftmost bit of the 5-bit field is the permanent file bit. If 1, the file is a direct access or indirect access with control PF. If the 5-bit field is 17, the LFN is either an indirect file without control or a non-permanent file type.

C = Completion bit. Set to 1 when function is completed.

Note

Due to the different concepts between the KRONOS 2 1 and SCOPE 3.4 permanent file structures, the following restrictions are in effect.

Any permanent file whose write lockout and/or execute bit(s) are set in the FNT, a permanent file with read only permission will be returned to the user.

26.2.8 DMP/REQ/CKP

SFP consists of routines which set up a special processing word in the calling control point area (SPCW) for follow-up processing by 1AJ and a CP program associated with the call.

Format of SPCW:

		6	18	18
SPCW =	ENTRY POINT	CODES	PARAMETER	PARAMETER ADDRESS

where:

ENTRY POINT = Name of entry in CP program.

CODES = Control codes for use by 1AJ 1/A, 1/B, 1/C, 1/D, 2/0

A = Request active (1AJ use only)
B = Clear RA-1 before reload if not set
C = Remainder of word is parameter list not address of parameter list
D = Do not restart CPU (1AJ use only)

PARAMETER = Input/output parameter

18/ parameter on input
12/ status, 6/ unused on output

PARM ADDR = Parameter address passed in call.

Each SFP routine will set the rollout flag in the control point area. The following dayfile message occurs:

SFP - SPECIAL REQUEST PROCESSING ERR. = the SPCW word was busy.

The three calls are:

CKP - Checkpoint request
REQ - Request equipment assignment
DMP - Dump field length

The KRONOS product sets are:

BASIC
APL
TSRUN
TEXT EDITOR
CYBERLINK
TRANEX
TELEX

All product sets are described in their respective Reference Manuals. The product sets are run as normal jobs with the exception of CYBERLINK, TRANEX, and TELEX which are subsystems. See Section 13 for a description of TELEX and Section 14 for a description of TRANEX.

CMR AND PP DUMPS 27

Octal dumps of central memory and PPU memory are available to the operator during dead start procedures. Some examples of these two types of dumps are given on the following pages. For a description of dead start dump selection consult the KRONOS 2.1 Operator's Guide, section 2.

Subsequent pages provide a partial listing of the CMR and PP 0, 1, 2, and 5 dumps. (Values from Figures 2-2 and 2-3.)

CPUMTR FWA 15050
LWA 16765

TPMN, TPPR FWA 16766

PPI EP 17142 from PXPP in CMR (63)

PPR - 1007 = 16057
PMN - 717 = 15767
MTR - 20 = 15070

BATCHIO at CP 26B, CPA at 5400B

PP Communication area - 6200

BATCHIO CPA at 5400 CP number 26

1st CPA at 200 CP number 1

EST at 6600

Notice that PP5 is 1TD and PP2 is 1TS.

MXD, SCX, SCX1, IXP start at 17042 from Figure 3-7.

27-2

000000	0000 0000 0000 0000 0000	0001 7646 0016 0014 3000	0003 3246 0000 0027 6200	6440 0001 4620 0000 0003
000004	6700 7700 0000 0001 0550	6600 6700 6601 0000 0000	0003 3245 0000 0000 0000	0003 3423 0003 4415 0000
000010	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000020	0000 0000 0000 0000 0350	2331 2324 0515 5555 0000	0000 0000 0001 0104 1700	0000 0000 0000 0000 0471
000024	0000 0010 1000 0010 0010	0000 0000 0000 0000 0000	0000 0000 0042 3635 4035	0000 0000 0311 1120 1326
000030	5534 4157 3434 5735 3557	5542 3650 3344 5033 4457	5501 2304 5555 1322 1716	1723 5526 3557 3455 5523
000034	5016 5534 3536 5500 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000040	0000 0000 0000 0000 0001	0000 0000 1413 0000 1406	0000 0000 0000 0002 0003	0001 1200 0000 0003 0002
000044	0000 0001 0000 0026 0025	0000 0027 0000 0000 0000	0003 3246 0000 0027 6200	0000 0000 0000 0000 6360
000050	0000 0000 0000 0242 5257	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000054	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	7777 7776 0000 1600 0000
000060	0000 0000 0000 0001 7122	7777 0000 0000 0000 0000	0000 0000 0060 0001 7142	0001 6957 0000 0000 0000
000064	6170 1073 0100 1073 7773	0000 2003 3016 6113 1073	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000070	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000074	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0400 0000 7600 0000 0000	0400 0000 7700 0000 0000
000100	0000 0000 0000 0000 0000	0000 0000 0000 4001 0005	0012 0000 0000 0000 0000	0100 4011 0000 0000 0000
000104	0000 0000 0000 0000 0000	0000 0000 0000 0100 0100	0000 1405 0000 0274 5771	0000 0000 0000 0000 0000
000110	0000 0000 0000 0000 0001	0000 0000 1450 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000114	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000200	0000 5254 0064 6300 0000	0003 5000 0000 0100 0001	0000 7700 0000 6200 0000	0007 0000 0056 4700 0033
000204	0000 0000 0001 2400 7214	0000 0000 0000 5600 0020	0000 0200 0000 0100 0040	0000 0000 0004 0700 0027
000210	0000 0000 0072 1400 0033	0000 0000 0000 0000 0000	0000 0000 0000 0000 0013	0000 0000 0000 0000 5730
000214	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	2203 1400 0000 0000 0000	0000 0000 0000 0000 0000
000220	2002 0000 0000 0350 0077	2405 1405 3055 5500 0000	0077 7775 0000 0000 0000	0000 1121 0000 0004 0000
000224	7777 7777 7700 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 2000	0000 0000 0000 0000 0000
000230	2605 2223 1117 1655 3557	3457 0000 0000 0000 0000	2405 1405 3055 1116 1124	1101 1411 3201 2411 1716
000234	5501 0217 2224 5700 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000240	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000250	0000 0000 0000 0001 5731	0000 0155 4200 0000 0655	0000 0000 0000 0000 0001	0000 0000 0000 0000 0000
000254	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000260	0500 0500 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000264	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	6705 4000 0001 0131 0134
000270	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	7777 7777 7777 7777 7777
000274	7777 7777 7777 7777 7777	7777 7777 7777 7777 7777	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000300	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000330	2405 1405 3057 0000 0000	2405 1405 3035 5755 0000	0530 1124 5755 0000 0000	2405 1405 3035 5755 0000
000334	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000400	0000 0665 0010 0000 0000	0007 7200 0000 0100 0001	0006 0000 0001 2300 0175	0007 0000 0000 0000 0000
000404	0000 0000 0000 0000 0003	0000 0000 0000 0200 0420	0000 0400 0000 0100 0001	0000 0000 0006 0600 0001
000410	7777 7777 7777 7700 0000	0000 0000 0000 0000 0000	5100 0000 0001 2300 0024	0213 2320 0000 0000 0000
000414	0000 0000 0000 0000 0000	1116 2025 2400 0000 0001	0516 0420 0000 0000 0000	1116 2025 2400 0000 0070
000420	0001 0000 0000 0772 0600	0411 2333 0103 2200 0000	0030 7760 0000 0000 0000	0000 1273 0000 0001 0000
000424	7777 7777 7700 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000430	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000450	0000 0000 0000 0000 0001	0000 0000 0100 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000454	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000460	0600 0600 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000464	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	6727 4000 0004 0135 0135
000470	4000 4306 4306 0001 0001	0101 0322 0000 0000 0000	0000 0000 0000 0000 0000	7777 7777 7777 7777 7777
000474	7777 7777 7777 7777 7777	7777 7777 7777 7777 7777	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000500	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000530	0411 2357 0000 0000 0000	1517 0405 5133 5200 0000	2205 2425 2216 5111 1620	2524 5200 0000 0000 0000
000534	0411 2357 0000 0000 0000	4035 4157 0000 0000 0000	0317 2031 0222 5111 1620	2524 5624 0523 2401 5255
000540	0000 0000 0000 0000 0000	2301 2605 5624 0523 2401	5755 0000 0000 0000 0000	0000 0000 0000 0000 0000
000544	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
000600	0000 0124 0004 5200 0000	0015 7200 0000 0102 4054	0002 5600 0165 6100 0435	0007 0000 0165 7177 7776

97404700A

000604	0000	0000	0017	3201	6567	0000	0000	0017	3301	6556	0000	0600	0000	0100	0000	0000	0000	0165	5600	0014
000610	0000	0000	0000	0002	4054	0000	0000	0000	0000	0000	0000	0000	0000	0001	6556	0000	0000	0000	0000	0012
000614	7777	7777	7777	7700	0000	0000	0000	0000	0001	6567	0102	2420	0000	0000	0000	1725	2420	2524	0000	0024
000620	0041	0000	0000	1572	0256	0101	0131	3335	3603	0000	0030	7000	0000	0000	0000	0000	1414	0000	0000	4077
000624	0000	0101	0000	0017	6750	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000630	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000634	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000650	0000	0000	0000	0000	0077	0000	0000	7700	0000	0012	0000	0000	0000	0000	0002	0000	0000	0000	0000	0000
000654	0000	0100	0000	0000	0077	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000660	0256	0256	0000	0000	0000	0000	0000	0000	0000	0000	0031	0000	0000	0000	0000	0000	0000	0000	0000	0557
000664	0217	0223	1115	3400	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6741	4000	0000	0130	0132
000670	0000	0000	0000	0000	0000	0101	0301	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0144	0024	0001
000674	0557	3700	0012	7777	1000	0000	0000	0000	0000	0215	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0400	0000	0000	0000	0000	0000	0000	0000	0000	0000
000704	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000730	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
000734	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001000	0000	0124	0004	5200	0000	0020	5000	0000	0102	4054	0002	5600	0165	6100	0435	0007	0000	0166	6777	7776
001004	0000	0000	0017	3201	6567	0000	0000	0017	3301	6556	0000	1000	0000	0100	0000	0000	0000	0165	5600	0014
001010	0002	4054	0004	3577	7776	0000	0000	0000	0000	0000	0000	0000	0000	0001	6556	0000	0000	0000	0000	0012
001014	7777	7777	7777	7700	0000	0000	0000	0165	6701	6556	0311	1720	0000	0001	6556	1725	2420	2524	0000	0024
001020	0041	0000	0000	2050	0256	0101	0131	3334	3403	0000	0030	7000	2011	0000	0000	0000	1414	0100	0000	4075
001024	0000	0100	0000	0017	5000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001030	5555	0000	0000	0000	0000	0200	0000	0061	0004	6000	5110	0160	7166	3006	3210	7660	0516	0007	1624	6000
001034	5160	0114	4171	6000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001050	0000	0000	0000	0000	0106	0000	0001	0600	0000	0012	0000	0000	0000	0000	0002	0000	0000	0000	0000	0000
001054	0000	0100	0000	0000	0013	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001060	0256	0256	0000	0000	0000	0000	0000	0000	0000	0000	0031	0000	0000	0001	6556	0000	0000	0000	0000	0557
001064	0217	0223	1115	3400	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6735	4000	0001	0132	0132
001070	0000	0000	0000	0000	0000	0101	0306	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0144	0024	0001
001074	0557	3700	0012	7777	1000	0000	0000	0000	0000	0215	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001130	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001134	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001200	0000	0124	0004	5200	0000	0023	2600	0000	0102	4054	0002	5600	0165	6100	0435	0007	0000	0165	7177	7776
001204	0000	0000	0017	3201	6567	0000	0000	0017	3301	6556	0000	1200	0000	0100	0000	0000	0000	0165	5600	0014
001210	0000	0000	0000	0002	4054	0000	0000	0000	0000	0000	0000	0000	0000	0001	6556	0000	0000	0000	0000	0012
001214	7777	7777	7777	7700	0000	0000	0000	0000	0001	6567	0102	2420	0000	0000	0000	1725	2420	2524	0000	0024
001220	0041	0000	0000	2326	0256	0101	0131	3335	3503	0000	0030	7000	0000	0000	0000	0000	1414	0000	0000	4102
001224	0000	0101	0000	0017	6750	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001230	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001234	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001250	0000	0000	0000	0000	0102	0000	0001	0200	0000	0012	0000	0000	0000	0000	0002	0000	0000	0000	0000	0000
001254	0000	0100	0000	0000	0102	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001260	0256	0256	0000	0000	0000	0000	0000	0000	0000	0000	0031	0000	0000	0000	0000	0000	0000	0000	0000	0557
001264	0217	0223	1115	3400	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6737	4000	0000	0130	0132
001270	0000	0000	0000	0000	0000	0101	0304	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0144	0024	0001
001274	0557	3700	0012	7777	1000	0000	0000	0000	0000	0215	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001300	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0400	0000	0000	0000	0000	0000	0000	0000	0000	0000
001304	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001330	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001334	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001400	0000	0243	0000	0000	0000	0026	0400	0010	6600	0020	0000	0000	0010	6700	0000	0007	0000	0000	0000	0000
001404	0000	0000	0010	7300	0000	0000	0000	0010	7400	0000	0000	1400	0011	4700	0000	0000	0000	0012	3300	0000

27-4

001410	4600	0460	0046	0004	6000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001414	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1147
001420	0000	0000	0000	2604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0154	0000	0001	0000
001424	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001430	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001464	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
001470	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
001474	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001600	0000	0362	6511	2700	0000	0026	0400	0004	2400	0000	0000	0000	0034	6400	0443	0007	0000	0000	0000	0000
001604	0000	0000	1271	5500	0423	0000	0000	3072	4453	4666	0000	1600	0365	0143	5223	0000	0000	0004	2464	7603
001610	4220	3072	4453	4665	4201	0000	0000	0000	0000	0000	0000	0000	6511	2700	0000	0000	0000	0000	0000	0000
001614	4220	3072	4453	4665	4201	0000	0000	0000	0077	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001620	0000	0000	0000	2604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0155	0000	0001	0000
001624	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001630	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0060	0000	0000	0800	0000	0000	0000
001664	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
001670	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
001674	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
001700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002000	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0037	0000	0000	0000	0000
002004	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2000	0000	0000	0000	0000	0000	0000	0000	0000
002010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002020	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002024	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002064	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
002070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
002074	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002200	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0007	0000	0000	0000	0000
002204	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2200	0000	0000	0000	0000	0000	0000	0000	0000
002210	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002220	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002224	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002230	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002264	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
002270	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
002274	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002300	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002400	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0007	0000	0000	0000	0000
002404	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2400	0000	0000	0000	0000	0000	0000	0000	0000
002410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002420	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002424	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002430	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002464	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
002470	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
002474	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002600	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0007	0000	0000	0000	0000

97404700A

002604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2600	0000	0000	0000	0000	0000	0000	0000	0000
002610	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002620	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002624	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002630	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002664	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000
002670	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
002674	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
002700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003000	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003004	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	3000	0000	0000	0000	0000	0000	0000	0000	0000
003010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003020	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003024	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003200	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003204	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	3200	0000	0000	0000	0000	0000	0000	0000	0000
003210	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003220	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003224	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003400	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003404	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	3400	0000	0000	0000	0000	0000	0000	0000	0000
003410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003420	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003424	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003600	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	3600	0000	0000	0000	0000	0000	0000	0000	0000
003610	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003620	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
003624	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004000	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004004	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4000	0000	0000	0000	0000	0000	0000	0000	0000
004010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004020	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004024	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004200	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004204	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4200	0000	0000	0000	0000	0000	0000	0000	0000
004210	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004220	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004224	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004400	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004404	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4400	0000	0000	0000	0000	0000	0000	0000	0000
004410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004420	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004424	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004600	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	4600	0000	0000	0000	0000	0000	0000	0000	0000
004610	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

5

27-6

	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004620	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
004624	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005000	0000	0000	0000	0000	0000	0026	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005004	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	5000	0000	0000	0000	0000	0000	0000	0000	0000
005010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005020	0000	0000	0000	2605	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005024	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005200	0000	0745	0000	0000	0000	0026	0500	0000	0100	0001	0000	1200	0002	3200	0000	0007	0000	0011	3300	0000
005204	0000	0000	0011	4000	0000	0000	0000	0000	3100	0000	0000	5200	0000	0100	0123	0000	0000	0002	3577	7776
005210	7777	7777	7777	7777	7760	2203	1400	0000	0000	0000	0000	0000	0000	0000	0000	2401	4017	4000	0000	0540
005214	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2203	1400	0000	0000	0000	0000	0000	0000	0000	0155
005220	2000	0000	0000	2605	0012	1501	0716	0524	0000	0000	0076	7772	0000	0000	0000	0000	0000	0000	0000	0000
005224	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005230	1501	0716	0524	5755	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005234	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005250	0000	0000	0000	0020	2511	0000	0000	0500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005254	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005260	0100	0100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005264	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6717	4000	0001	0131	0134
005270	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
005274	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0010	0000	3100	7700	0041	0000	0000	0000	0300	0000
005300	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005330	1501	0716	0524	5755	0000	1501	0716	0524	3457	0000	0530	1124	5755	0000	0000	1501	0716	0524	3457	0000
005334	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005400	0000	0000	0000	0000	0000	0026	1700	0000	0000	0000	0000	1100	0000	0000	0000	0007	0000	0000	0000	0000
005404	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	5400	0000	0000	0000	0000	0000	0000	0000	0000
005410	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005420	0001	0000	0000	2617	0011	0201	2403	1011	1700	0000	0001	7773	0000	0000	0000	0000	0000	0000	0000	0000
005424	7777	7777	7700	0000	0000	3411	1766	0001	0002	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005430	5534	5502	2506	0605	2223	5501	0324	1126	0557	0000	0100	1132	1400	3436	1405	0000	0000	0000	0000	0000
005434	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005450	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0170	0000	0000	0000	0000	0000
005454	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005464	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6715	4000	0000	0000	0000
005470	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
005474	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005500	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005530	2414	2033	0104	0611	0000	0004	0012	0104	0611	0000	0000	0000	0000	0012	0000	0000	0000	0000	0000	0000
005534	0000	0000	0000	0020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0021	0000	0000	0000	0000	0000	0000
005540	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005600	0001	5764	0004	4100	0000	0027	7500	0062	2300	0001	0000	0300	0062	2400	0002	0007	0000	0062	2100	5771
005604	0000	0000	0062	2200	0000	0000	0000	0002	1000	0001	0000	5600	0000	0101	5754	0000	0000	0002	3177	7775
005610	0000	0000	0015	7540	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005614	0000	0000	0000	0000	0000	2414	3020	0000	0001	5765	2414	3020	0000	0001	5765	0000	0000	0000	0000	0000
005620	0001	0000	0000	2775	0003	2324	1115	2514	0100	0000	0076	7770	0000	0000	0000	0000	0000	0000	0000	0000
005624	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005630	2324	1115	2514	0124	1117	1655	1116	1124	1101	2405	0457	0000	0000	0000	0000	0000	0000	0000	0000	0000
005634	0000	0000	0000	0000	0000	0000	0000	0000	0000	0020	1523	0700	0002	0000	3713	0000	0000	0000	0000	0000
005640	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005650	0000	0000	0000	0000	3142	0000	0031	4200	0000	0267	0000	0070	0300	0000	0002	0000	0000	0000	0000	0000
005654	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

97404700A

6

005660	0100	0100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005664	0000	0000	0000	0000	0002	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	6713	4000	0001	0131	0131
005670	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	7777	7777
005674	7777	7777	7777	7777	7777	7777	7777	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0060	0000	0000	0000	0000	0000	0000
005730	2324	1115	2514	0157	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
005734	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006000	0001	6427	0000	0000	0000	0000	0000	0150	5200	0001	0030	0000	0077	7300	0000	0000	0000	0077	7177	7747
006004	0000	0000	0077	0100	7773	0000	0000	0063	7177	7723	0000	6000	0150	5077	7717	0000	0000	0063	7100	0004
006010	0000	4000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7777	7777	0000	0000	0000	0000	0274	0000
006014	0000	0000	0000	0000	4200	7777	7777	7777	7777	7760	0000	0000	0000	0000	0001	0000	0000	0000	0274	4244
006020	0002	0000	0000	0000	3000	2331	2324	0515	5555	0000	0100	7777	0000	0000	0000	0000	0000	0000	0000	0000
006024	7777	7777	7700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006030	0000	0000	0000	0000	0000	5504	0501	0455	2324	0122	2455	1405	2605	1455	5533	5700	0000	0000	0000	0000
006034	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006050	0000	0000	0000	0000	1305	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006054	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006200	1524	2225	0000	0000	0000	0000	0000	0000	0000	0000	3411	1766	0000	0003	0001	0000	0000	0000	0000	0000
006204	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006210	0423	0430	0000	0000	0000	0000	6330	0000	0000	0000	3404	1430	0010	0044	0124	0000	0000	0000	0000	0000
006214	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006220	3424	2327	0000	0000	0002	0000	0000	0000	0000	0000	0020	1405	0000	5753	0020	1655	1116	1124	1101	2405
006224	0457	0000	0000	0000	0000	5755	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006230	4747	4700	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006234	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006240	3424	1701	0033	0000	0127	0012	0020	0000	0411	0770	1725	2420	2524	0000	0604	7042	0020	0000	0411	0770
006244	0000	0000	0000	0000	0000	5700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006250	3424	0441	0031	0020	0011	0000	0000	0000	0000	4000	0000	5770	0000	0003	0002	3074	1660	6210	1454	0200
006254	0364	3001	0100	7422	0000	2357	0000	5513	5755	1116	0100	1522	3654	1003	2100	0004	6020	3024	2177	3436
006260	4747	4700	0006	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006264	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006270	4747	4700	0007	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006274	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006300	0000	0000	0000	0000	6420	0000	0000	0000	0000	0000	0000	0000	0411	0153	0153	7042	0020	0000	0411	0770
006304	5700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006310	3401	1205	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0411	0153	0153	7042	0020	0000	0411	0770
006314	5700	0000	0000	0000	0000	1355	2331	2324	0515	3057	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006320	3403	0426	0001	0000	0000	0000	0012	0000	2617	0011	0311	1726	0000	0000	0030	7042	0020	0000	0411	0770
006324	5700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006330	3401	1203	0000	0000	0000	0022	0000	0165	7177	7776	0201	2311	0353	0000	0000	7042	0020	0000	0411	0770
006334	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006340	1735	4102	0000	0000	0010	0003	0010	0000	0000	0000	0000	0000	0411	0153	0153	7042	0020	0000	0411	0770
006344	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006350	4747	4700	0023	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006354	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006360	0000	0000	0000	0000	6300	0000	0000	0000	0000	0000	2001	0000	0024	0022	0017	7042	0020	0000	0411	0770
006364	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006370	3422	1704	0000	0000	0000	0012	0020	0000	0001	0550	0000	0000	0411	0153	0153	7042	0020	0000	0411	0770
006374	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006400	4747	4700	0026	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006404	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006410	4747	4700	0027	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006414	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006420	0000	0000	0000	0000	6430	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006424	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006440	0001	3520	0207	0400	0000	0000	4206	4206	0011	0000	0001	0620	0002	0100	0000	0000	0000	0000	0000	0000
006444	0001	0720	0011	0100	0000	0000	0000	0000	0000	0000	0001	1020	0003	0100	0000	0000	0000	0000	0000	0000
006450	0001	1120	0011	0100	0000	0000	0000	0000	0000	0000	0001	1220	0003	0100	0000	0000	0000	0000	0000	0000
006454	0001	1320	0000	0100	0000	0000	0000	0000	0000	0000	0001	1420	0000	0100	0000	0000	0000	0000	0000	0000

7

27-8

006460	0001	1520	0000	0100	0000	0000	0000	0000	0000	0000	0001	1620	0000	0100	0000	0000	0000	0000	0000	0000
006464	0001	1720	0000	0100	0000	0000	0000	0000	0000	0000	0001	2020	0000	0100	0000	0000	0000	0000	0000	0000
006470	0001	2120	0000	0100	0000	0000	0000	0000	0000	0000	0001	2220	0000	0100	0000	0000	0000	0000	0000	0000
006474	0001	2320	0000	0100	0000	0000	0000	0000	0000	0000	0001	2420	0000	0100	0000	0000	0000	0000	0000	0000
006500	0001	2520	0000	0100	0000	0000	0000	0000	0000	0000	0001	2620	0000	0100	0000	0000	0000	0000	0000	0000
006504	0001	2720	0000	0100	0000	0000	0000	0000	0000	0000	0001	3020	0000	0100	0000	0000	0000	0000	0000	0000
006510	0001	3120	0000	0100	0000	0000	0000	0000	0000	0000	0001	3220	0002	0100	0000	0000	0000	0000	0000	0000
006514	0001	3320	0000	0100	0000	0000	0000	0000	0000	0000	0001	3420	0006	0100	0000	0000	0000	0000	0000	0000
006520	0001	4120	0032	0400	0000	0000	4207	4207	0011	0000	0001	4520	0006	0100	0000	0000	4210	4210	0001	0000
006524	0020	0021	0001	0150	0164	0303	0011	0100	0135	0023	0077	0353	0003	0224	0143	0271	0000	0234	0143	0210
006530	0143	0271	0000	0054	0145	0274	0031	0066	0066	0061	0000	0077	0077	0077	0077	0025	0025	0025	0025	0077
006534	0077	0052	0022	0070	0045	0022	0050	0070	0045	0027	0076	0077	0077	0025	0025	0000	0003	0002	0010	0006
006540	0004	0010	0004	0012	0004	0014	0004	0002	0001	0150	0001	0000	0207	0021	0000	0115	0000	0017	0000	0003
006544	0000	0022	0000	0025	0377	0023	0006	0007	0012	0002	0001	0014	0000	0016	0001	0064	0001	0017	0000	0162
006550	0133	0036	0000	0127	0001	0043	0004	0022	0000	0127	0002	0010	0005	0041	0004	0023	0011	0007	0016	0002
006554	0001	0162	0236	0014	0200	0017	0001	0162	0115	0024	0006	0017	0001	0162	0112	0025	0006	0024	0002	0014
006560	0200	0015	0001	0064	0001	0162	0056	0010	0001	0041	0004	0023	0012	0017	0001	0162	0077	0014	0200	0024
006564	0006	0017	0001	0162	0073	0026	0006	0024	0002	0014	0200	0015	0001	0064	0001	0162	0044	0010	0001	0041
006570	0004	0023	0012	0027	0005	0023	0037	0007	0042	0002	0001	0162	0173	0105	0000	0001	0154	0000	0207	0021
006574	0010	0017	0001	0162	0047	0021	0002	0162	0017	0010	0001	0041	0004	0023	0006	0021	0004	0017	0001	0162
006600	7000	0020	0000	0411	0770	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006604	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006610	0002	0010	0000	0423	7000	0026	0012	0000	0322	4000	0000	0012	0000	4320	5000	0000	0000	0000	0000	0000
006614	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006620	0000	0012	0000	1421	6000	0000	0012	0000	1420	3000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006624	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006630	2000	0004	0020	6424	7000	2001	0011	0020	2424	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006634	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006640	0000	0004	0000	6324	7000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006644	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006650	0000	0013	0000	1524	6000	0000	0013	0000	1524	6001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006654	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006660	0000	0013	0000	1524	5000	0000	0013	0000	1524	5001	0000	0013	0000	1624	5002	0000	0013	0000	1624	5003
006664	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006674	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2405	0000	2000	0000	0000	1605	0000
006700	2331	2324	0515	0001	0700	0000	4002	4002	0001	0005	2601	1411	0425	3000	1300	0000	4242	0000	0000	0005
006704	2301	1426	0122	0500	0701	0000	4236	4236	0001	0305	2205	2305	3004	0600	1300	0000	4245	0000	0000	0005
006710	2205	2305	3026	0600	1300	0000	4246	0000	0000	0005	3424	2327	0102	2000	0627	0000	0000	0000	0000	0001
006714	3411	1726	0101	1600	0626	0000	0000	0000	0000	0001	3415	2425	0101	1700	0625	0000	0000	0000	0000	0001
006720	2324	1115	0614	0000	0627	0000	4243	4243	0003	0015	1605	2726	0114	0000	0700	0000	4237	4237	0001	0505
006724	2601	1400	0000	0000	0700	0000	4241	4241	0001	0505	1116	2025	2447	0001	0002	0000	4306	4306	0002	0005
006730	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	2414	2033	0104	0600	0026	0000	4307	4307	0027	0307
006734	2405	2324	0200	0000	1104	0000	4303	4303	0003	0005	2405	2324	0200	0000	1105	0000	4271	4271	0001	0005
006740	2405	2324	0200	0000	1103	0000	4277	4277	0001	0005	1725	2420	2524	0000	0605	0031	0000	0000	0000	0005
006744	1725	2420	2524	0000	0603	0031	0000	0000	0000	0005	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
006750	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
007700	0044	7054	0624	0000	2604	7777	0153	0001	5300	0153	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
007704	4234	4000	4235	0020	4211	7042	0000	0000	0003	0001	0100	0000	0000	0001	0377	0000	0000	0000	0000	0000
007710	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
007720	4213	3777	4003	4004	4017	4005	4006	4007	4010	0017	4011	4012	4013	4014	0017	4015	4016	4017	4020	0017
007724	4021	4022	4023	4024	0017	4025	4026	4027	4030	0017	4031	4032	4033	4034	0017	4035	4036	4037	4040	0017
007730	4041	4042	4043	4044	0017	4045	4046	4047	4050	0017	4051	4052	4053	4054	0017	4055	4056	4057	4060	0017
007734	4061	4062	4063	4064	0017	4065	4066	4067	4070	0017	4071	4072	4073	4074	0017	4075	4076	4077	4100	0017
007740	4101	4102	4103	4104	0017	4105	4106	4107	4110	0017	4111	4112	4113	4114	0017	4115	4116	4117	4120	0017
007744	4121	4122	4123	4124	0017	4125	4126	4127	4130	0017	4131	4132	4133	4134	0017	4135	4136	4137	4140	0017
007750	4141	4142	4143	4144	0017	4145	4146	4147	4150	0017	4151	4152	4153	4154	0017	4155	4156	4157	4160	0017
007754	4161	4162	4163	4164	0017	4165	4166	4167	4170	0017	4171	4172	4173	4174	0017	4175	4176	4177	4200	0017
007760	0057	4202	4203	4204	0017	4205	0103	0011	0011	0017	0001	4212	0017	4214	0017	4215	4216	4217	4220	0017
007764	4221	4222	4223	4224	0017	4225	4226	4227	4230	0017	4231	4232	4233	0000	0017	0012	0001	0002	0004	6017
007770	0044	0044	0005	0003	5017	4256	0002	0002	0006	3017	4314	0006	0006	4264	0017	0006	4312	0000	0006	0017
007774	0006	0000	0006	0000	0012	0111	4313	0003	0004	0017	0004	0003	0003	0003	0017	0004	0004	0003	0003	0017

97404700A

8

97404700A

010000	0004	0003	0005	0503	0017	0003	0004	0002	0027	0017	0006	0036	0111	0111	0017	0111	0000	0000	0000	0010
010004	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010120	0000	3777	0000	0000	0034	0000	0000	0000	0000	0000	0003	0000	0000	0000	0000	0000	0000	0000	0000	0000
010124	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010544	0241	0000	0100	0141	0010	0021	0353	0006	0010	0027	0030	0000	0343	0002	0100	0164	0241	0001	0100	0141
010550	6600	0700	3000	0001	0000	6000	0100	1000	0001	0000	0400	0100	7700	0001	0000	0031	0100	0020	0000	0000
010554	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010560	2400	2000	4010	0001	0000	2400	1010	4004	0001	0000	0200	0100	7300	0001	0000	0030	0400	0200	0000	0000
010564	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010570	3400	2400	4010	0001	0000	3400	1400	4006	0001	0000	0200	0100	7600	0001	0000	0030	0400	0200	0000	0000
010574	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010600	4000	3770	7006	0001	0000	4004	3740	7000	0001	0100	0200	0100	7000	0001	0000	0030	0040	0010	0000	0000
010604	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010610	6774	6700	7400	0001	0000	6774	4000	7400	0001	0000	6000	0100	7700	0001	0000	0031	0400	0060	0000	0000
010614	7777	7777	7777	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
010620	5534	4157	3342	5734	4357	2405	1405	3057	0000	0000	3356	2442	4242	4257	0000	5534	4157	3341	5734	3757
010624	0103	0317	2516	2456	1514	1756	5700	0000	0000	0000	5534	4157	3341	5734	3757	0317	1520	0123	2357	0000
010630	5534	4157	3341	5734	4257	5501	2323	0515	0214	3155	0317	1520	1405	2405	5755	5555	3737	3333	3302	5523
010634	0315	5525	2305	0457	0000	5534	4157	3341	5734	4257	5555	5555	3457	4342	3455	5503	2025	5523	0503	1716
010640	0423	5555	0123	2305	1502	1431	5524	1115	0557	0000	5534	4157	3341	5734	4257	0320	5555	5555	5555	5534
010644	5743	4235	5523	0503	5700	0000	0000	0000	0000	0000	5534	4157	3341	5734	4257	0315	5555	5555	5555	5533
010650	5733	3435	5513	2710	5700	0000	0000	0000	0000	0000	5534	4157	3341	5734	4257	1523	5555	5555	5555	5533
010654	5733	4234	5513	2022	5700	0000	0000	0000	0000	0000	0405	0611	1605	5626	0114	1104	2530	5700	0000	0000
010660	5534	4157	3334	5735	4357	0317	2031	5616	0527	2601	1456	2601	1411	0425	3056	2657	0000	0000	0000	0000
010664	5534	4157	3334	5735	4457	5505	1604	5517	0655	1116	0617	2215	0124	1117	1655	0516	0317	2516	2405	2205
010670	0457	0000	0000	0000	0000	5534	4157	3334	5735	4457	5526	0522	1106	3155	0717	1704	5700	0000	0000	0000
010674	5534	4157	3334	5737	3557	0317	2031	5626	0114	5626	0114	1116	0430	5626	5700	0000	0000	0000	0000	0000
010700	5534	4157	3334	5737	3557	5505	1604	5517	0655	1116	0617	2215	0124	1117	1655	0516	0317	2516	2405	2205
010704	0457	0000	0000	0000	0000	5534	4157	3334	5737	3657	5526	0522	1106	3155	0717	1704	5700	0000	0000	0000
010710	5534	4157	3334	5737	4157	0320	5555	5555	5555	5533	5733	4335	5523	0503	5700	0000	0000	0000	0000	0000
010714	5534	4157	3334	5737	4157	1523	5555	5555	5555	5533	5734	4035	5513	2022	5700	0000	0000	0000	0000	0000
010720	5534	4157	3344	5740	3557	0411	2357	0000	0000	0000	5534	4157	3344	5740	3557	1517	0405	5133	5200	0000
010724	5534	4157	3344	5740	3557	2205	2425	2216	5111	1620	2524	5200	0000	0000	0000	5534	4157	3344	5740	3557
010730	0411	2357	0000	0000	0000	0303	1725	1624	5006	0115	1114	3157	0000	0000	0000	5534	4157	3344	5740	3657
010734	0320	5555	5555	5555	5533	5733	3334	5523	0503	5700	0000	0000	0000	0000	0000	2301	2605	5624	0523	2402
010740	5755	0000	0000	0000	0000	5534	4157	3340	5740	3457	0321	5555	5555	5555	5533	5733	3342	5523	0503	5700
010744	0000	0000	0000	0000	0000	5534	4157	3340	5740	3457	1523	5555	5555	5555	5533	5733	3337	5513	2022	5700
010750	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011020	5534	4157	3433	5734	3557	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000
011024	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011120	5534	4157	3433	5735	3557	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	5534	4157	3433	5735	3557
011124	2405	2324	0255	5555	4646	3333	4133	3333	0000	0000	5534	4157	3433	5740	4357	5314	0403	5617	5624	0523
011130	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011134	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011220	5534	4157	3433	5734	4357	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000	0000	0000	0000	0000	0000
011224	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011320	5534	4157	3333	5733	3557	2201	1657	0000	0000	0000	5534	4157	3333	5733	3557	1517	0405	5133	5200	0000
011324	5534	4157	3333	5733	3557	2201	1657	0000	0000	0000	5534	4157	3333	5733	3757	5517	2005	2201	2417	2255
011330	0422	1720	5700	0000	0000	5534	4157	3333	5733	3757	0320	5555	5555	5555	5533	5735	4140	5523	0503	5700
011334	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
011420	5534	4157	3333	5733	3557	0114	2357	0000	0000	0000	5534	4157	3333	5733	3557	1517	0405	5133	5200	0000
011424	5534	4157	3333	5733	3557	0114	2357	0000	0000	0000	5534	4157	3333	5733	3757	5517	2005	2201	2417	2255
011430	0422	1720	5700	0000	0000	5534	4157	3333	5733	3757	0320	5555	5555	5555	5533	5735	3440	5523	0503	5700
011434	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
013220	5534	4157	3333	5733	4257	1501	0716	0524	5755	0000	5534	4157	3333	5733	3757	5517	2005	2201	2417	2255
013224	0422	1720	5700	0000	0000	5534	4157	3333	5733	3757	0530	1124	5755	0000	0000	5534	4157	3333	5733	3757

27-9

27-10

013230	1501 0716 0524 3457 0000	5534 4157 3333 5733 3757	5555 5555 3442 5741 3434	5520 0522 0305 1624 5503
013234	2025 5525 2411 1411 3201	2411 1716 5700 0000 0000	5534 4157 3333 5733 3757	5527 0111 2455 3415 2455
013240	0317 1520 1405 2405 5700	0000 0000 0000 0000 0000	5534 4157 3333 5733 3757	5515 0107 1605 2455 2405
013244	2215 1116 0124 0550 1617	5524 0120 0555 1217 0223	5700 0000 0000 0000 0000	5534 4157 3333 5733 3757
013250	0320 5555 5555 5555 5533	5740 3643 5523 0503 5700	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
013254	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0030 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
013320	5534 4157 3333 5733 3757	5517 2005 2201 2417 2255	0422 1720 5700 0000 0000	0000 0000 0000 0000 0000
013324	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
013420	5534 4157 3342 5735 3657	2324 1115 2514 0157 0000	5534 4157 3343 5734 4257	2324 1115 2514 0124 1117
013424	1655 1116 1124 1101 2405	0457 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
013430	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000	0000 0000 0000 0000 0000
013520	5534 4157 3433 5733 4357	0101 0101 3333 3515 5755	1523 1722 2456 3333 3340	5633 3335 3357 0000 0000
013524	5534 4157 3433 5733 4457	0101 0131 3335 3324 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013530	5534 4157 3433 5733 4457	0101 0131 3333 4024 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013534	5534 4157 3433 5734 3357	0101 0101 3333 3515 5755	1523 1722 2456 3333 3536	5633 3334 4057 0000 0000
013540	5534 4157 3433 5734 3457	0101 0131 3334 4024 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013544	5534 4157 3433 5734 3557	0101 0131 3335 3624 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013550	5534 4157 3433 5734 3657	0101 0101 3333 3515 5755	1523 1722 2456 3333 3435	5633 3333 4157 0000 0000
013554	5534 4157 3433 5734 3757	0101 0131 3333 4124 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013560	5534 4157 3433 5734 4057	0101 0131 3334 3524 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000
013564	5534 4157 3433 5734 4157	0101 0101 3333 3515 5755	1523 1722 2456 3333 3337	5633 3335 3556 3333 3442
013570	5700 0000 0000 0000 0000	5534 4157 3433 5734 4257	0101 0131 3334 4224 5755	5314 0403 5617 5624 0523
013574	2402 5656 3457 0000 0000	5534 4157 3433 5734 4357	0101 0131 3335 3524 5755	5314 0403 5617 5624 0523
013600	2402 5656 3457 0000 0000	5534 4157 3433 5734 4357	0101 0131 3333 3724 5755	5314 0403 5617 5624 0523
013604	2402 5656 3457 0000 0000	5534 4157 3433 5735 3457	0101 0101 3333 3515 5755	1523 1722 2456 3333 3436
013610	5633 3333 4256 3333 3434	5700 0000 0000 0000 0000	5534 4157 3433 5735 3557	0101 0131 3334 3424 5755
013614	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3433 5735 3557	0101 0131 3334 3424 5755
013620	2405 2324 0255 5555 4646	3333 4133 3333 0000 0000	5534 4157 3433 5735 3557	0101 0131 3333 4224 5755
013624	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3433 5735 3557	0101 0131 3334 3624 5755
013630	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3433 5735 4457	0101 0101 3333 3515 5755
013634	1523 1722 2456 3333 3534	5700 0000 0000 0000 0000	5534 4157 3433 5736 3357	0101 0131 3335 3424 5755
013640	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3433 5736 3357	0101 0131 3335 3424 5755
013644	2405 2324 0255 5555 4646	3333 4133 3333 0000 0000	5534 4157 3433 5737 4357	0101 0101 3333 3515 5755
013650	1523 1722 2456 3333 3441	5700 0000 0100 0000 0000	5534 4157 3433 5737 4457	0101 0131 3334 4124 5755
013654	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3433 5740 4257	0101 0101 3333 3515 5755
013660	1523 1722 2456 3333 3434	5700 0000 0000 0000 0000	5534 4157 3433 5740 4357	0101 0131 3334 3424 5755
013664	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3434 5733 4157	0101 0101 3333 3515 5755
013670	1523 1722 2456 3333 3433	5700 0000 0000 0000 0000	5534 4157 3434 5733 4257	0101 0131 3334 3324 5755
013674	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3434 5733 4257	0101 0101 3333 3515 5755
013700	1523 1722 2456 3333 3534	5700 0000 0000 0000 0000	5534 4157 3434 5733 4457	0101 0131 3335 3424 5755
013704	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3434 5734 3457	0101 0101 3333 3515 5755
013710	1523 1722 2456 3333 3437	5700 0000 0000 0000 0000	5534 4157 3434 5734 3557	0101 0131 3334 3724 5755
013714	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	5534 4157 3434 5735 3357	0101 0101 3333 3515 5755
013720	1523 1722 2456 3333 3536	5633 3335 3556 3333 3440	5700 0000 0000 0000 0000	5534 4157 3434 5735 3457
013724	0101 0131 3334 4024 5755	5314 0403 5617 5624 0523	2402 5656 3457 0000 0000	2324 1115 0102 1102 5755
013730	0103 0317 2516 2456 0217	0223 1115 3556 5700 0000	5534 4157 3340 5740 3657	2324 1115 0102 1102 5755
013734	5511 1414 0507 0114 5501	0303 1725 1624 5006 0115	1114 3157 0000 0000 0000	5534 4157 3341 5734 3757
013740	2414 2033 0102 1402 5755	2414 2056 0315 4133 3333	3356 2442 4242 4257 0000	5534 4157 3341 5734 3757
013744	2414 2033 0102 1402 5755	0103 0317 2516 2456 1514	1756 5700 0000 0000 0000	5534 4157 3341 5734 3757
013750	2414 2033 0102 1402 5755	0317 1520 0123 2357 0003	5534 4157 3341 5734 4257	2414 2033 0102 1402 5755
013754	5501 2323 0515 0214 3155	0317 1520 1405 2405 5755	5555 3737 3333 3302 5523	0315 5525 2305 0457 0000
013760	5534 4157 3341 5734 4257	2414 2033 0102 1402 5755	5555 5555 3457 4342 3455	5503 2025 5523 0503 1716
013764	0423 5555 0123 2305 1502	1431 5524 1115 0557 0000	5534 4157 3342 5734 4357	2405 1405 3055 5523 5755
013770	2405 1405 3057 0000 0000	5534 4157 3342 5735 3657	2324 1115 2514 0123 5755	2324 1115 2514 0157 0000
013774	5534 4157 3343 5734 4257	2324 1115 2514 0123 5755	2324 1115 2514 0124 1117	1655 1116 1124 1101 2405
014000	0457 0000 0000 0000 0000	5534 4157 3344 5736 3557	0101 0101 3333 3515 5755	1523 1722 2456 3333 3437
014004	5700 0000 0000 0000 0000	5534 4157 3344 5736 3757	0101 0101 3333 3515 5755	1523 1722 2456 3333 3441
014010	5700 0000 0000 0000 0000	5534 4157 3344 5737 3457	0101 0101 3333 3515 5755	1523 1722 2456 3333 3533
014014	5700 0000 0000 0000 0000	5534 4157 3344 5737 3657	0101 0101 3333 3515 5755	1523 1722 2456 3333 3433
014020	5633 3334 4057 0000 0000	5534 4157 3344 5737 4057	0101 0101 3333 3515 5755	1523 1722 2456 3333 3536
014024	5633 3335 3557 0000 0000	5534 4157 3344 5737 4257	0101 0101 3333 3515 5755	1523 1722 2456 3333 3341

97404700A

97404700A

10

014030	5633	3333	4057	0000	0000	5534	4157	3344	5737	4457	0101	0101	3333	3515	5755	1523	1722	2456	3333	3435
014034	5633	3333	3756	3333	3442	5700	0000	0000	0000	0000	5534	4157	3344	5740	3557	0411	2333	0103	2223	5755
014040	0411	2357	0000	0000	0000	5534	4157	3344	5740	3557	0411	2333	0103	2223	5755	1517	6405	5133	5200	0000
014044	5534	4157	3344	5740	3557	0411	2333	0103	2223	5755	2205	2425	2216	5111	1620	2524	5200	0000	0000	0000
014050	5534	4157	3344	5740	3557	0411	2333	0103	2223	5755	0411	2357	0000	0000	0000	5534	4157	3344	5740	3657
014054	0101	0101	3333	3515	5755	1523	1722	2456	3333	3436	5633	3334	3456	3333	3342	5700	0000	0000	0000	0000
014060	5534	4157	3344	5740	4257	0101	0101	3333	3515	5755	1523	1722	2456	3333	3437	5700	0000	0000	0000	0000
014064	5534	4157	3344	5740	4357	0101	0131	3334	3724	5755	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000
014070	5534	4157	3433	5733	3557	0101	0101	3333	3515	5755	1523	1722	2456	3333	3534	5700	0000	0000	0000	0000
014074	5534	4157	3433	5733	3757	0101	0101	3333	3515	5755	1523	1722	2456	3333	3441	5700	0000	0000	0000	0000
014100	5534	4157	3433	5733	4057	0101	0131	3334	4124	5755	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000
014104	5534	4157	3433	5733	4057	0101	0131	3334	4124	5755	2405	2324	0255	5555	4646	3333	4133	3333	0000	0000
014110	5534	4157	3433	5733	4057	0101	0101	3333	3515	5755	1523	1722	2456	3333	3433	5700	0000	0000	0000	0000
014114	5534	4157	3433	5733	4157	0101	0131	3334	3324	5755	5314	0403	5617	5624	0523	2402	5656	3457	0000	0000
014120	0000	0000	0000	0000	0000	5534	4157	3344	5733	4157	0133	2531	3334	3324	5755	2006	5555	5555	5555	5507
014124	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5734	3357	0133	2531	3334	3524	5755
014130	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5734	3457
014134	0133	2531	3334	3424	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000
014140	5534	4157	3344	5734	3457	0133	2531	3335	3424	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257
014144	0000	0000	0000	0000	0000	5534	4157	3344	5734	4057	0133	2531	3334	3624	5755	2006	5555	5555	5555	5507
014150	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	2324	1115	0102	0702	5755	1523	5555	5555	5555	5533
014154	5733	3337	5513	2022	5700	0000	0000	0000	0000	0000	5534	4157	3340	5740	3357	2324	1115	0102	1023	5755
014160	0322	5555	5555	5555	5533	5733	3737	5513	0304	5755	0000	0000	0000	0000	0000	5534	4157	3340	5740	3457
014164	2324	1115	0102	1002	5755	2006	5555	5555	5555	2301	2605	5555	2405	2324	0257	0000	0000	0000	0000	0000
014170	5534	4157	3340	5740	3457	2324	1115	0102	1002	5755	0320	5555	5555	5555	5533	5733	3342	5523	0503	5700
014174	0000	0000	0000	0000	0000	5534	4157	3340	5740	3457	2324	1115	0102	1002	5755	1523	5555	5555	5555	5533
014200	5733	3337	5513	2022	5700	0000	0000	0000	0000	0000	5534	4157	3340	5740	3557	2324	1115	0102	0712	5755
014204	1420	5555	5555	5555	5533	5733	3734	5513	1416	5755	0000	0000	0000	0000	0000	5534	4157	3340	5740	3657
014210	2324	1115	0102	1123	5755	0322	5555	5555	5555	5533	5733	3736	5513	0304	5755	0000	0000	0000	0000	0000
014214	5534	4157	3340	5740	3657	2324	1115	0102	1102	5755	0320	5555	5555	5555	5533	5733	3334	5523	0503	5700
014220	0000	0000	0000	0000	0000	5534	4157	3340	5740	3757	2324	1115	0102	1002	5755	1420	5555	5555	5555	5533
014224	5733	3734	5513	1416	5755	0000	0000	0000	0000	0000	5534	4157	3340	5740	4257	2324	1115	0102	1102	5755
014230	1420	5555	5555	5555	5533	5733	3643	5513	1416	5755	0000	0000	0000	0000	0000	5534	4157	3341	5734	3757
014234	2414	2033	0102	1423	5755	0322	5555	5555	5555	5533	5736	3444	5513	0304	5755	0000	0000	0000	0000	0000
014240	5534	4157	3341	5734	4257	2414	2033	0102	1402	5755	0320	5555	5555	5555	5534	5743	4235	5523	0503	5700
014244	0000	0000	0000	0000	0000	5534	4157	3341	5734	4257	2414	2033	0102	1402	5755	0315	5555	5555	5555	5533
014250	5733	3435	5513	2710	5700	0000	0000	0000	0000	0000	5534	4157	3341	5734	4257	2414	2033	0102	1402	5755
014254	1523	5555	5555	5555	5533	5733	4234	5513	2022	5700	0000	0000	0000	0000	0000	5534	4157	3341	5740	3457
014260	2414	2033	0102	1402	5755	1420	5555	5555	5555	5533	5737	4243	5513	1416	5755	0000	0000	0000	0000	0000
014264	5534	4157	3342	5734	4457	2405	1405	3055	5523	5755	2006	5555	5555	0124	2401	0310	5555	2311	1506	1114
014270	0555	2331	2324	0515	3057	0000	0000	0000	0000	0000	5534	4157	3342	5734	4457	2405	1405	3055	5523	5755
014274	2006	5555	5555	0124	2401	0310	5555	1605	2427	1722	1355	2331	2324	0515	3057	0000	0000	0000	0000	0000
014300	5534	4157	3343	5733	3757	2324	1115	2514	0123	5755	2006	5555	5555	5555	5507	0524	5555	2324	1115	0614
014304	5755	0000	0000	0000	0000	5534	4157	3343	5733	3757	2324	1115	2514	0123	5755	2006	5555	5555	5555	5507
014310	0524	5555	2324	1115	0614	5755	0000	0000	0000	0000	5534	4157	3343	5736	4157	0101	0131	3333	3724	5755
014314	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5736	4357
014320	0101	0131	3334	4224	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000
014324	5534	4157	3343	5737	3457	0101	0131	3334	3724	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705
014330	0455	1116	5700	0000	0000	5534	4157	3343	5737	3457	0101	0131	3334	4124	5755	2523	0522	5555	0217	0223
014334	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	3557	0101	0131	3333	4224	5755
014340	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	3657
014344	0101	0131	3334	3324	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000
014350	5534	4157	3343	5737	3657	0101	0131	3333	4124	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705
014354	0455	1116	5700	0000	0000	5534	4157	3343	5737	3657	0101	0131	3334	4024	5755	2523	0522	5555	0217	0223
014360	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	3757	0101	0131	3335	3624	5755
014364	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	3757
014370	0101	0131	3335	3324	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000
014374	5534	4157	3343	5737	4057	0101	0131	3333	4024	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705
014400	0455	1116	5700	0000	0000	5534	4157	3343	5737	4257	0101	0131	3335	3524	5755	2523	0522	5555	0217	0223
014404	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	4357	0101	0131	3334	3524	5755
014410	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3343	5737	4457
014414	0101	0131	3334	3424	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000
014420	5534	4157	3343	5737	4457	0101	0131	3334	3624	5755	2523	0522	5555	0217	0223	1115	3455	5514	1707	0705
014424	0455	1116	5700	0000	0000	5534	4157	3343	5740	3457	0101	0131	3335	3424	5755	2523	0522	5555	0217	0223

27-11

27-12

II

014430	1115	3455	5514	1707	0705	0455	1116	5700	0000	0000	5534	4157	3344	5733	3357	0133	2531	3334	4124	5755
014434	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5733	3357
014440	0133	2531	3333	3724	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000
014444	5534	4157	3344	5733	3457	0133	2531	3333	4124	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257
014450	0000	0000	0000	0000	0000	5534	4157	3344	5733	3457	0133	2531	3335	3524	5755	2006	5555	5555	5555	5507
014454	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5733	3557	0133	2531	3334	4024	5755
014460	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5733	3657
014464	0133	2531	3334	3724	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000
014470	5534	4157	3344	5733	3657	0133	2531	3334	4224	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257
014474	0000	0000	0000	0000	0000	5534	4157	3344	5733	3757	0133	2531	3335	3324	5755	2006	5555	5555	5555	5507
014500	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5733	4057	0133	2531	3333	4224	5755
014504	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000	5534	4157	3344	5733	4057
014510	0133	2531	3335	3624	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257	0000	0000	0000	0000	0000
014514	5534	4157	3344	5733	4157	0133	2531	3333	4024	5755	2006	5555	5555	5555	5507	0524	5555	2405	2324	0257
014520	5533	3357	3334	5734	4257	2331	2324	0515	5555	5755	5542	3650	3342	5033	4057	5504	0501	0455	2324	0122
014524	2455	1405	2605	1455	5533	5700	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
014530	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
014620	4603	6323	7000	0020	0000	4107	0004	0000	6440	0225	0000	0000	5053	0000	2550	0000	1100	0000	0000	0000
014624	0411	0153	0153	0100	1017	0100	6160	0100	1011	0100	3302	0324	2400	0100	6130	5400	0633	0200	0702	5600
014630	0716	7720	0004	2000	0502	7420	7120	7000	0520	1404	0322	5400	0653	0200	0702	5600	0716	7720	0005	2000
014634	0502	7420	7320	0030	0517	6620	0655	7520	3402	7720	0012	7420	7020	5400	0573	0410	2200	2012	0527	2042
014640	0411	0200	0533	0102	0611	0100	0623	5600	0711	1400	5400	0575	2000	0003	2300	0153	0405	2000	0004	3307
014644	0460	1500	3412	3410	3007	5400	0716	3413	3610	3013	2177	7624	0672	2000	0000	3310	0405	7720	0010	6420
014650	0743	7720	0002	3010	5400	0736	5100	0577	3410	3013	5400	0711	3513	3066	2200	3776	1020	3411	1056	3113
014654	3413	3612	3013	1730	0673	3006	1021	0603	1411	3512	7420	1404	7320	0010	1470	0100	0655	1502	5400	0716
014660	1400	0100	0605	1420	0200	0364	3077	6170	0576	6010	2000	1043	34 1	0303	4502	3601	4001	3402	3011	1720
014664	0571	0100	0544	0625	0631	0632	0645	0651	0652	0655	0657	0661	0663	0664	0741	0743	0745	1002	1004	1037
014670	0212	0004	0100	0017	0200	0200	0200	2403	2300	1073	0000	0614	0100	1157	0200	4574	3052	1104	0503	0100
014674	1250	0200	3217	5000	4573	0503	0100	1243	1405	3401	3074	1630	63 1	4573	0200	4353	0534	0200	2366	0425
014700	0736	0200	1414	0200	3131	3057	0503	0200	1507	0200	1630	0200	2175	0200	3751	0505	2036	0103	0200	0533
014704	2000	4421	0100	3013	1400	5400	1645	0200	2366	0466	0712	5000	1732	5400	1635	0200	3131	0200	1630	0354
014710	5000	3004	1016	0705	2001	4573	0200	0501	3046	0415	0200	0501	1403	3411	1440	0200	0364	3074	1667	6240
014714	0100	0103	3065	0410	1110	0511	3411	1462	0200	0364	0362	3074	1667	6240	0100	1367	0200	3777	5000	2515
014720	0200	4353	0521	0200	2366	3053	1071	0427	3057	0525	0200	4315	0422	1440	3523	3053	1071	1102	0414	2000
014724	4573	3401	1400	4401	3601	2300	4650	0572	6020	0326	3074	2100	0100	6170	3004	5000	3004	1014	0715	3077
014730	6220	0200	2442	3077	6020	3002	3424	0200	3455	5500	1350	3055	1006	1664	6220	1410	3401	0205	2646	6301
014734	4573	3053	1071	0514	3074	1667	6240	5000	3004	1016	0705	2000	4573	0200	0501	1402	3411	1462	0200	0364
014740	0100	0103	1014	3450	1071	3351	1377	3351	3451	3075	6250	0100	0103	5600	1505	0100	1137	3074	1650	3401
014744	1614	6010	3013	1014	0764	3022	2300	0500	0562	3021	2300	1115	0556	3020	5400	1471	2300	0324	0407	2300
014750	2100	0545	2000	0106	3401	3001	6010	3077	6210	1400	3411	1450	0200	0364	3077	6170	1474	2001	1471	0100
014754	1212	0324	1115	0557	3333	3333	3333	5733	3333	5523	0503	5700	0000	0000	0100	1506	0200	4315	0473	3057
014760	6032	1701	6010	3013	1017	0714	3052	1102	0411	1101	0407	3045	1103	0404	1400	0100	3420	3032	1277	3405
014764	5100	0551	6010	3010	1006	0664	3034	0505	2000	4435	0100	3013	3057	1701	6010	3014	1071	1100	0405	3033
014770	3434	1401	3435	3034	3406	3035	3407	3074	1661	6025	0100	1734	5000	1505	0403	0100	3114	1407	6006	0200
014774	3366	0416	3010	1014	3111	1701	6060	0100	1734	1400	6010	3074	2100	0101	6210	0100	1147	1405	3457	0200
015000	6020	0442	6060	1701	6325	1602	6010	0200	2664	0305	5000	3004	1014	0655	3030	1020	0703	5600	2515	3063
015004	1006	0704	1401	0100	3420	3060	0417	5400	3300	1006	0613	3060	2200	3777	5400	3300	3256	0604	3056	5400
015010	3300	3074	1661	6025	3010	0423	1071	0521	3025	1377	3310	3425	3011	3426	3012	3427	3013	3430	3074	1661
015014	6225	0303	0471	0463	0200	3276	5900	3004	1015	0603	0100	2024	3054	1201	1013	3525	3024	0410	3067	3466
015020	0200	2517	3054	1301	3454	3052	1102	0504	2000	2000	3525	3055	1006	1664	6220	1614	6225	3057	1710	0733
015024	0200	0547	3004	0200	0437	0200	5712	0423	3401	3004	0200	0446	3001	1102	0503	0100	1550	1400	0100	3420
015030	1400	6010	3055	1006	1664	6210	0200	5064	0403	0100	3013	3013	1237	1014	3314	0200	3670	0200	2442	3065
015034	1703	0632	0200	3455	5500	2077	1410	3401	3055	1006	1664	6020	3012	3424	3023	1377	3423	3055	1006	1664
015040	6220	1604	6301	4573	5000	3004	1017	0607	2036	0106	0200	0533	0200	5652	3074	2100	0100	6370	3004	6025
015044	3402	5000	3004	1015	0606	3026	1302	3426	3002	6225	3074	1621	6025	1646	6240	3031	0413	1277	3411	1405
015050	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0274	5761
015054	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	7777	7600	0000	0000	0000	0000	0000	0000	0000	0000
015060	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	3423	1230	0000	0000	0000	3415	0120	0000	0000	0000
015064	1717	0004	0611	1564	5707	2000	0000	0000	0000	1750	5075	0000	0061	0004	6000	0132	0000	0061	0004	6000
015070	5622	1212	4453	5216	3320	0302	0150	7603	1501	5317	5612	0211	4462	3177	7775	0630	0150	6751	1200	0003
015074	4321	1717	0000	0021	5612	5461	0020	0015	2574	6000	5110	0150	5003	0101	5067	5120	0150	5213	7775	4710
015100	0312	0150	6771	0000	0000	0400	0151	7361	0004	6000	5321	0206	3771	3000	7600	7372	0114	3676	2102	0635
015104	5234	0000	2020	2605	4710	5261	0000	0036	7325	4730	0233	0000	0061	0044	6000	5627	0563	7121	2445	4431

97404700A

97404700A

12

015110	0302	0151	3021	4443	7624	2134	4032	6015	1303	6732	4300	6522	7000	0001	1602	5343	1122	6403	0201	5130
015114	4330	3766	7015	1134	3001	5020	0000	6012	7102	1607	6336	0212	3354	7102	0060	6362	0511	0015	0552	3530
015120	5137	0000	2226	6434	6000	5146	0000	2226	6541	5315	0645	0151	2412	7155	4710	0400	0150	6761	0004	6000
015124	2120	7633	2023	4301	2734	7607	0613	0015	1321	0500	5471	0056	0015	1641	0633	5461	0040	0015	1644	6000
015130	7170	0000	0704	0001	5257	7157	0000	0071	0700	0000	5040	0000	6061	4041	5135	2044	4522	4000	0202	0203
015134	0332	0152	2361	0004	6000	2043	0137	0563	7502	0530	6364	0126	5767	4626	3200	5060	0000	6003	0701	5142
015140	0305	0151	4251	1700	0020	4370	1207	7112	6175	4610	0440	0150	7077	5644	3316	5716	4522	5000	0011	0611
015144	7140	0030	0061	5000	0002	1072	2546	1454	7242	0302	4600	0541	1554	2251	0611	0333	0151	4510	7225	4614
015150	5472	4204	5271	1000	0000	7661	0127	4574	3502	0322	5365	1201	5553	7554	3600	5466	5127	1354	6615	4761
015154	5066	0000	0202	0001	5070	6130	0151	5604	0001	5164	5117	0000	5151	2700	0050	4363	6157	1621	1363	7621
015160	2710	6214	0627	3044	2613	5130	0150	6440	6632	6406	3617	4202	3612	7215	4710	0244	0000	0061	0054	6000
015164	5010	0000	6051	2000	0106	4363	0157	2650	3001	5253	2113	0522	1000	0243	7373	5211	0000	5015	2263	6613
015170	5473	0372	6203	2201	5172	5461	0023	3000	0004	6000	2760	6546	1002	3300	0000	5117	0000	2051	2001	5055
015174	4360	3154	1676	7701	2604	2170	7760	1050	3000	0060	2005	7633	7023	1301	5721	5461	0213	3054	7206	3430
015200	0547	0150	6703	0701	5215	5422	1133	3326	1301	5672	6170	0000	0761	5000	0022	2740	6135	5504	0001	5210
015204	7333	6226	7353	1652	6641	0634	0152	1072	6177	7775	7410	0136	61 3	0601	5210	1053	3613	4000	0004	6000
015210	1514	0244	6103	1401	5204	7630	0111	2727	4017	6200	0311	0152	1063	3502	3130	1567	1205	0755	6211	0055
015214	6130	0151	3203	1501	5164	7060	0000	0013	5552	0604	7206	0171	2261	3001	5132	0400	0151	6461	0004	6000
015220	5120	0150	5251	1000	6020	1262	0546	2003	3161	5067	6170	0060	00 4	6001	5114	5234	0000	2043	7015	3141
015224	2077	1436	3015	7375	6221	1136	1201	3054	7306	3610	5214	0000	0437	7231	1661	5122	0000	0456	7213	7626
015230	5462	0133	4061	3001	5232	6353	0020	0015	1644	6000	5234	0000	1252	1400	0016	2630	3762	0026	1313	7363
015234	0450	0152	4005	3001	5240	2060	1613	0000	0012	6703	0336	0152	4051	2600	0001	2072	2372	2761	3000	0000
015240	2660	3523	4000	0172	7731	5463	0557	3163	7401	0622	6634	0566	6161	6000	0000	0333	0153	1043	7301	5337
015244	0200	0153	1061	0004	6000	4376	0721	0770	0051	3207	6660	0032	1015	2506	6610	7211	0000	0503	3101	5255
015250	0302	0152	5520	1022	2211	3611	2512	6000	0446	3610	2326	2157	2720	7074	3160	5227	0000	2221	2441	3002
015254	1520	1030	2015	2551	3777	5120	0150	5113	0271	5001	0233	0000	0061	0004	6000	5117	0000	2043	2142	0103
015260	0321	0152	6451	3001	5267	6337	0226	3303	2601	5264	5117	0000	1627	7315	4710	0200	0151	3161	0004	6000
015264	2026	0201	7115	3122	0744	1263	7546	1071	0000	0000	0200	0151	7361	0004	6000	3020	0460	0061	0004	6000
015270	5146	0000	0251	3700	0020	2004	4502	0000	0603	6640	2002	2367	3021	2305	4640	2144	4633	2051	7700	0020
015274	0537	0152	7756	2602	1644	7170	0000	0221	2443	7326	0302	0152	5733	2301	5257	6140	0153	0003	1401	5155
015300	5120	0000	4103	2201	5303	5512	1767	1020	7141	2617	5562	1040	0015	0674	6000	5140	0150	6151	1000	0047
015304	0314	0150	6751	3001	5062	4370	1030	1015	0672	2603	1277	2613	0015	0675	4720	0400	0151	0261	0004	6000
015310	5117	0000	2043	6362	0103	6157	0000	1703	3301	5314	5120	0150	6427	3034	0632	2660	6263	0320	6441	2663
015314	0331	0153	1561	5700	0024	5127	0000	5056	6502	6602	5462	0023	3000	0004	6000	5137	0000	2054	1212	0303
015320	0333	0151	3120	5222	1144	7375	0205	5262	4762	5470	0440	0155	0506	4001	5334	6257	7466	6004	5001	5412
015324	0650	0153	2762	6776	7553	0460	0154	1562	4776	7676	0640	0153	4634	0001	5414	6257	6373	5504	5001	5456
015330	0650	0154	6062	4774	5762	6257	7261	7304	4001	5420	0450	0154	4462	4774	6360	0440	0154	1002	0001	5346
015334	6247	5574	6362	5753	6662	0440	0155	3704	5001	5655	0650	0153	4207	4001	5532	6247	5554	7504	4001	5551
015340	0640	0156	0462	4755	7163	0440	0155	4602	0001	5346	6247	4765	5762	5753	6347	0440	0157	1204	5001	5677
015344	6247	4765	5504	4001	5743	6257	4476	7606	5001	5414	5117	0000	2243	2602	0130	1561	2763	1072	1677	0016
015350	0321	0153	5120	3501	2553	5127	0000	2051	1000	0047	2022	3713	0007	6001	1632	7276	7767	7751	2700	0022
015354	0327	0153	6403	0101	5364	2024	3205	3703	3201	5364	1545	3762	7012	6427	6700	2063	5507	5000	0002	0560
015360	6130	0150	6703	2501	5102	7601	0737	1120	0515	4750	1266	0613	0015	1732	0020	0400	0151	0261	0004	6000
015364	7601	0200	7204	0001	5173	4375	3150	2766	6207	3510	0300	0154	0072	5000	0077	0335	0154	1421	5062	0506
015370	3705	1202	5203	2001	5403	5147	0000	2051	3700	0061	4360	5206	6511	7644	3660	0307	0153	7651	4700	0022
015374	2043	0157	4672	4777	0017	0334	0153	6461	0004	6000	2030	3166	2371	7777	2177	2167	4110	6003	0701	5403
015400	1377	7547	5004	5001	5270	2053	6767	1012	6575	6635	0400	0152	7061	0004	6000	0300	0154	0051	2001	5063
015404	0450	0154	0671	7000	0001	2053	6126	5756	6352	0536	1252	5734	1061	4001	5346	0400	0151	5561	0004	6000
015410	5110	0154	1371	6003	1117	1111	5206	5212	5164	6000	0200	0153	5161	0004	6000	0000	0077	0000	0077	7777
015414	7170	0000	0504	0001	5257	4311	4512	7000	0202	0160	7170	0000	0415	3212	0744	3663	7546	2004	0001	5444
015420	5120	0154	2743	3601	0455	7305	0204	4415	7436	3470	3710	1232	4272	7777	7703	4762	2114	1722	3166	3430
015424	2027	3116	2467	5041	3777	0326	0153	4650	7500	0000	0255	0154	4461	0004	6000	0000	0000	0014	1404	0020
015430	5117	0000	6543	2521	1321	3663	0546	1002	0001	5067	5117	0000	0310	6112	1666	5360	3020	0015	0674	6000
015434	5117	0001	0210	6115	3603	0200	0150	6761	0004	6000	5310	3106	1151	6700	0102	0200	0150	6761	0004	6000
015440	5117	0000	6443	2521	1621	5360	3020	0015	0674	6000	5117	0000	6010	6115	3603	0200	0150	6761	0004	6000
015444	4341	1512	7000	0202	0471	5137	0000	2511	6427	6000	1233	6031	3015	1734	6000	0200	0151	7361	0004	6000
015450	7647	0511	0000	0474	6000	7130	3401	1220	4357	6510	0301	0151	7320	5651	2725	6130	0151	7320	3524	6000
015454	5072	0000	0012	6344	6000	0400	0151	0261	0004	6000	7110	0000	0120	1501	2551	0400	0153	4661	0004	6000
015460	6247	6360	7362	5762	7262	0440	0154	6604	5001	5476	6267	6373	5105	6001	5346	6355	0045	0015	4667	6110
015464	7160	0000	0420	6521	3256	2015	0125	2104	0001	5346	5110	0154	7354	2111	0511	2260	2543	2151	4200	0002
015470	1073	3214	2253	6135	4761	1064	4511	7000	0612	2701	5662	0537	2304	0001	5346	1404	2220	0000	0000	0010
015474	1404	2254	0000	0000	0100	0000	0140	0000	0000	0000	6355	0634	1006	1501	5414	0654	0154	1420	5515	6235
015500	2123	6032	5015	3652	0523	0312	0153	4651	3200	0005	4303	6767	1021	3061	1603	1267	6437	0056	6355	4750
015504	0400	0150	6761	0004	6000	6355	0432	6006	1501	5414	6341	0215	3006	5401	5414	1545	2626	4777	7725	6335

27-13

13

27-14

015510	4361	4117	6303	1701	5513	7661	0127	3654	7302	1336	6353	0563	3506	5401	5414	0660	0155	1502	6601	5521
015514	7621	0040	0015	5304	6000	5443	1040	0015	5214	6000	5443	1040	0015	5264	6000	7621	1040	0015	5304	6000
015520	7120	0000	0304	0001	5530	1063	3227	0454	3415	4431	5167	0000	3054	7611	0633	2270	4543	4154	6715	4761
015524	1162	3437	0054	6715	6731	0400	0150	6761	0004	6000	1063	3227	0451	6760	0035	5476	1543	4104	0001	5524
015530	7675	0202	3051	1001	5063	1222	7365	1204	0001	5346	6247	6057	7262	6757	7163	0440	0154	1005	6001	5346
015534	5632	0434	3073	2502	0552	1533	4737	5120	7441	2637	6650	0566	2004	0001	5365	2052	3760	1013	7772	0072
015540	0335	0155	4251	7300	0001	0400	0151	7361	0004	6000	2055	1634	5306	1401	5414	5623	4635	1006	4501	5414
015544	2027	3032	2015	1735	6731	0400	0150	6761	0004	6000	7325	0215	2253	5566	3410	0450	0153	6506	1501	5414
015550	0754	0153	6504	0001	5414	5147	0001	0043	6605	3353	6563	3204	1162	5100	0000	0324	0153	4606	1601	5414
015554	6343	0205	5206	6501	5414	2132	2730	5007	4101	5414	7343	0213	2203	0301	5414	6363	4767	3406	6501	5414
015560	6130	0155	7162	4377	7676	0640	0154	1463	4306	3570	0310	0152	4551	2000	0001	4330	0157	2620	7065	6150
015564	0331	0155	7461	0004	6000	7251	7776	4703	2501	5414	5321	7106	2254	6166	7441	4320	1044	0015	6005	4111
015570	0311	0155	6504	0001	5600	0337	0154	1403	0701	5602	0300	0156	0352	2700	0020	1572	6202	6015	3262	0706
015574	7364	4203	0637	6676	3330	0326	0154	1463	3436	7441	5613	4106	1156	6546	7441	0640	0155	7643	2014	6000
015600	5443	0436	0012	7245	4650	5473	0020	0015	0674	6000	4320	1202	7314	0001	5600	7601	0200	7202	0001	5173
015604	6247	5466	7405	4001	5346	5127	0001	0051	3700	0075	4366	0514	7000	0222	0211	2036	2122	2320	4305	3353
015610	1574	6724	7770	0161	6224	0322	-0153	4620	5526	5633	6251	0000	0053	4532	0336	0616	0154	1463	4401	5036
015614	0665	0154	1463	6542	1314	0604	0154	1406	6501	5414	6130	0156	1702	0001	5245	0337	0154	1471	1477	7676
015620	7223	7777	7503	2101	5414	0322	0154	1473	3306	6510	0307	0156	4552	1700	0076	0300	0156	4652	2700	0020
015624	4600	0031	3015	6252	0136	1572	6202	6073	0102	1122	1532	6207	0615	1162	0306	6361	0731	0403	0001	5643
015630	0764	0156	4437	7712	0222	3600	3033	7015	6515	3100	0332	0156	4603	1101	5646	4370	0106	4452	7000	0000
015634	0441	0156	3761	0004	6000	5444	1674	4110	7445	4771	0714	0156	3561	0004	6000	5260	0000	0061	0004	6000
015640	5443	0767	5043	0367	6600	1140	4546	5012	7475	4730	0200	0150	6761	0004	5000	6150	0000	1102	0001	5640
015644	6150	0000	0702	0001	5640	6150	0000	0302	0001	5640	7601	0205	4561	5000	0005	0325	0156	4020	0724	6000
015650	0200	0151	7361	0004	6000	5443	0716	0000	0034	3036	7670	0114	0454	7501	2646	6072	7777	5750	6300	0000
015654	7170	0000	1202	0001	5257	6355	0432	6006	1501	5414	6341	0215	3006	5401	5414	1545	2626	4777	7704	6000
015660	0660	0156	6102	6601	5670	6130	0156	7304	0001	5164	5110	0000	3104	0001	5670	5110	0000	3004	0001	5670
015664	5110	0000	2602	0001	5670	5110	0001	0602	0001	5672	5110	0001	0604	0001	5670	5110	0000	2704	0001	5670
015670	1061	1437	0056	6355	6731	0400	0150	6761	0004	6000	4323	0156	1261	0004	6000	5120	0150	6465	3512	7106
015674	4262	1407	2151	3001	5065	2650	7404	3620	5141	2154	0400	0156	7061	0004	6000	5117	0000	2243	6602	0130
015700	1561	6723	6770	0165	3453	0333	0153	4610	5442	0437	7130	0076	0720	5221	5643	7215	4345	4472	5576	7676
015704	0321	0154	1476	2701	2662	5110	0000	4703	3501	5414	2063	5137	7703	0101	5711	5474	0613	0015	0675	4750
015710	0400	0151	0261	0004	6000	5475	0040	0015	0674	6000	6355	0624	5000	0206	3610	0615	0154	1407	6401	5414
015714	5435	5544	3143	0301	1404	5023	0000	0303	0401	5414	3663	4544	2121	6441	1702	3761	6114	0403	3601	5414
015720	3677	4511	2000	0057	6630	3771	7206	4403	3701	5414	3663	6213	4450	1177	7776	5465	5111	0176	0353	6712
015724	2055	2557	4127	6036	3630	6130	0157	2751	6200	0016	0716	0151	6402	0011	5414	5117	0000	2443	7521	5357
015730	4323	0227	0115	1125	4441	3636	3517	2000	0171	5332	4323	0371	1361	3001	5132	7657	0516	2000	0121	5742
015734	0331	0154	1461	0004	6000	7440	0204	0472	6401	7362	2064	4127	6750	7400	0000	7110	0150	7074	2002	0144
015740	7167	0000	0020	2221	2712	5264	0170	7152	7401	7762	0200	0153	1061	0004	6000	6355	0432	6006	1501	5414
015744	2054	4616	5000	0206	3410	1565	2066	4015	4146	3460	6266	7777	7506	6001	5414	0244	0157	5061	0004	6000
015750	5110	0157	6602	0001	5752	6653	5137	7702	0001	5760	5623	5434	3065	4221	3777	2260	2567	3102	0001	5756
015754	1522	4460	0011	3433	6632	5562	4201	0154	2211	0622	0321	0157	5563	6105	5324	0660	0157	5402	0001	5067
015760	5612	0562	2143	3166	7452	5673	1106	1161	5000	0002	1072	2546	1454	7242	0302	4600	0541	1554	2251	0611
015764	0333	0157	6210	7225	4614	5472	4020	0015	0674	6000	3360	0200	0000	0000	0002	0310	0157	7350	1000	0060
015770	2113	0562	2121	2446	3710	0302	0150	7653	5216	3320	0315	0153	1704	0001	5067	7210	7777	6303	2101	5776
015774	5230	0167	6663	6302	0052	6370	0200	2202	6600	0000	5350	0261	4561	5400	1741	0400	0160	6061	0004	6000
016000	5010	0000	6061	3001	5067	2113	0637	1004	7001	5164	5622	0563	2121	2442	1344	3673	2430	0653	2701	1602
016004	5343	1122	6403	1201	5164	7170	0000	0704	0001	5257	5117	0000	2020	1025	6271	0321	0150	6721	2445	3521
016010	4332	2114	3503	0401	5067	6130	0160	1204	0001	5164	5627	1653	5104	0001	5317	7601	0200	7204	0001	5220
016014	2106	0561	7151	2700	0020	2015	2366	1020	0363	6720	5461	0547	2?04	0001	5067	5010	0000	6066	6202	1060
016020	6351	0200	0604	5701	5270	7460	0763	1013	6636	6670	5216	0000	6021	1304	6000	6251	0000	0005	5701	5270
016024	0400	0150	6761	0004	6000	4330	3114	3051	1700	0020	1521	3366	4254	6107	6770	0320	0150	6721	7077	6210
016030	5110	0150	5563	4702	0257	2334	2126	1351	6001	5355	0400	0150	6761	0004	5000	5110	0160	3510	6114	6000
016034	5160	0160	5704	0001	5067	0400	0150	6761	0004	6000	5110	0160	4010	6114	6000	5160	0160	5704	0001	5067
016040	5455	0261	4561	5400	1741	5617	0211	4403	0101	6044	5622	0212	4403	1201	5067	5010	0000	6021	1306	3710
016044	7170	0000	0204	0001	5257	7637	0213	0751	2001	5055	4370	1624	3000	0132	3647	6634	1222	3251	6001	5056
016050	0322	0150	6751	1700	0020	0331	0151	0704	0001	5067	5010	0150	5724	3317	6410	2045	7030	1015	0672	3334
016054	7623	0137	1354	7102	0203	5252	0062	0126	1454	6000	6154	0017	4104	0001	6060	5455	0261	4561	5400	1741
016060	7174	0017	0107	5001	6436	5525	1032	7016	4062	0237	7140	0076	0051	3501	7002	1164	2437	0063	6306	3760
016064	0266	0000	0061	0004	6000	7110	0000	0366	6002	0144	0400	0161	0561	0004	6000	1071	1211	3511	5412	0151
016070	5235	0300	2115	2243	6625	5245	0000	2076	2102	0635	0313	0160	73?3	2101	5066	2040	6513	0000	0574	3014
016074	0334	0150	6620	5601	1303	2010	2370	5343	3017	6500	2026	0030	0316	1034	5000	2017	3117	3155	6511	2447
016100	5137	0000	2020	4663	6742	5474	0376	3210	7555	4630	0200	0150	6661	0004	5000	0331	0150	6671	5000	0011
016104	2056	0020	0016	0774	6000	5147	0000	2151	2700	0020	0304	0164	0643	3602	0203	2034	4151	1303	3201	6112

97404700A

016110	2027	1112	3236	6215	4620	0301	0150	6604	0001	6120	0301	0161	1054	7502	1144	1071	1020	0015	2574	6000
016114	2112	2624	1777	7564	6000	6251	7777	7606	4001	6414	5375	5675	5106	5001	6116	0200	0150	6661	0004	6000
016120	5117	0000	2051	2001	5055	0321	0161	2576	6702	1607	6236	0000	1450	3000	0060	2263	2033	6016	1252	1330
016124	6243	0000	0005	4701	6417	0560	0161	3076	2102	0260	5130	0000	4710	6333	7712	5565	1547	1075	6515	4630
016130	4320	3116	2176	0002	0660	5465	0040	0015	1734	6000	7671	0512	0000	0412	0744	0332	0150	6651	1001	5061
016134	7601	0137	7761	3001	5066	0400	0164	2161	0004	6000	5147	0000	2251	1700	0020	6254	7777	7606	5001	6150
016140	7621	0513	0000	0472	0260	1063	3377	1251	4700	0025	4351	4113	5720	2051	2334	7631	0532	1003	0301	6153
016144	1017	7556	5154	7104	3202	7565	1546	3013	7772	0273	1142	1031	4016	2444	6000	0400	0150	6661	0004	6000
016150	6040	0000	0004	4501	6140	7160	0000	4420	6605	4650	0400	0164	1761	0004	6000	1261	2714	0340	1127	6370
016154	5461	0204	5276	7002	0335	1264	3556	5104	0001	5066	6430	0433	4451	0000	0000	1505	3201	2703	2101	6162
016160	0110	0001	0104	0001	6267	5603	0547	5004	0001	5067	0120	0001	0104	0001	6164	5603	0547	5004	0001	5067
016164	4331	4203	6012	7305	6030	5475	0040	0015	0674	6000	2114	4513	7000	0207	6510	6241	7777	7220	5654	6000
016170	0640	0164	0602	4401	6176	1563	5546	3004	0001	5066	1533	5765	0004	0001	6176	1563	5546	3004	0001	6136
016174	1533	5765	1004	0001	6176	5117	0000	2104	0001	6210	7627	0212	0652	1200	6440	4341	4204	4415	6145	4711
016200	5461	0517	7000	0351	3666	5167	0000	3051	7700	0025	5167	0000	2251	7700	0071	5167	0000	2143	4302	0460
016204	1563	4030	5016	2077	5751	5140	0000	4720	5605	4740	3766	5107	4455	7514	6000	5463	0137	7704	0001	5066
016210	5147	0000	2251	2001	6237	0301	0150	6610	6442	2103	1233	5225	0251	2700	0025	2064	3032	6016	2154	3510
016214	7620	0205	7061	0004	6000	1165	1511	0000	0471	2626	4350	3030	1015	0661	1553	0316	0150	6676	6702	0544
016220	0323	0162	2451	2001	5055	2160	7623	6000	0142	2732	5020	0000	6021	2306	3420	0337	0162	2405	4701	6417
016224	7621	0124	4520	2301	1724	7150	3401	1220	7031	2647	7677	0205	5254	6402	0735	5341	0126	5720	2303	6732
016230	5464	0547	3010	6442	1771	5461	0030	7015	0661	3777	5117	0000	2171	2077	0000	1142	1547	5072	6474	7777
016234	1300	0031	6015	1734	3560	1543	5614	0015	1732	0406	0400	0151	5561	0004	6000	7777	0000	7777	0000	0000
016240	5117	0000	2043	2021	1621	0316	0162	4204	0001	5066	5147	0000	2251	1700	0020	6254	7777	7606	5001	6261
016244	5120	0162	6311	6215	4750	0316	0162	5151	2001	5055	0321	0151	0776	4702	1407	6234	0000	1422	2324	6000
016250	0332	0151	0704	0001	5067	2060	2562	7103	2601	5067	2124	4523	2050	0017	4450	1363	4030	6016	2604	6000
016254	7120	2203	1427	6022	0652	1326	3212	4403	1201	5067	4320	3763	1015	1122	0372	1261	3546	1004	0001	5067
016260	5073	0000	0004	0001	5107	6040	0000	0004	4501	6244	0470	0162	4434	0001	6417	1000	7777	0000	0000	0000
016264	6430	0432	4415	0525	4051	0110	0000	0104	0001	6267	5603	0547	5004	0001	5067	6150	0162	6602	0001	6377
016270	5110	0000	4754	2511	0622	0301	0150	6661	3001	6273	0400	0151	0261	0004	6000	2014	4107	1104	0001	5066
016274	5475	0130	0051	1001	5061	6130	0153	0004	0001	6421	4334	4211	3061	3001	5066	1761	3030	6015	3102	7101
016300	5120	0150	6542	3122	6303	0230	0153	1061	0004	6000	5321	0614	0010	5504	3073	6361	0116	0220	2737	6710
016304	0664	0164	0661	4000	6700	2074	4032	2015	0663	6777	6150	0077	0007	6401	6312	0756	0163	1252	3177	7776
016310	5045	0000	0143	2061	3534	2022	2154	5203	1401	5066	5462	0137	7712	0001	5066	5130	0150	5261	5001	5067
016314	0313	0164	1402	0001	6660	2113	0613	0016	3204	3360	1501	3512	0015	0512	1114	6351	0734	2002	0001	5245
016320	5227	0000	7671	6000	0004	0337	0164	0603	0701	6340	7160	0000	0203	0201	6344	1314	7766	1003	0101	6340
016324	5227	0000	2015	0232	1214	1542	3200	0672	2577	7776	2054	4215	6636	1256	3350	2040	6370	2063	6416	3426
016330	7675	0032	0016	4066	4550	6660	0030	7016	3355	6240	6754	1604	5000	0016	6661	7160	0000	0303	1201	6340
016334	5634	0107	3304	3101	6337	5614	6106	1166	6615	6656	0763	0163	3513	0761	5066	5675	1137	6602	0001	5066
016340	2064	4107	6604	0001	5066	4346	0513	7000	0532	0144	1571	4710	0007	7723	6637	1154	1613	0016	3455	4630
016344	2053	0020	0015	2454	6000	0307	0164	0655	2517	3170	5237	0000	2010	7552	0135	0300	0150	6663	5502	0237
016350	7140	0076	0004	5001	6067	4306	0767	1011	6032	1544	5137	0000	2020	6666	3350	6366	3207	3037	3371	0633
016354	2033	0033	3016	4065	4630	1150	6157	6056	3602	0566	6343	0207	0663	3701	3777	5335	4567	6006	1401	6406
016360	0643	0160	6776	0103	6630	0551	0163	6350	6300	0000	0200	0160	6761	0004	5000	5433	1435	0620	5301	1753
016364	4355	3031	7016	3654	3562	7605	0205	0115	0054	3754	2070	2113	7636	6305	5631	0200	0160	6761	0004	6000
016370	2114	4542	5103	1101	6372	7621	0711	0000	0202	0230	5331	7366	3250	6300	0000	0200	0161	3661	0004	6000
016374	6430	0432	4415	0525	4051	0120	0000	0161	0004	6000	5603	0547	5014	0001	5067	4371	4616	0000	1012	0760
016400	7120	0000	0151	0001	6405	0110	0000	0104	0001	6404	6766	1360	0205	6001	6401	0255	0000	0061	0004	6000
016404	1277	0025	5000	0004	6000	0000	0000	0000	0000	0000	6150	0150	6761	0004	5000	5110	0000	2751	2001	6413
016410	1061	1227	0250	6500	0006	5170	0060	3571	6001	0000	5160	0060	2602	5500	0000	2020	5510	2516	0757	0000
016414	7035	7715	7727	1077	6210	2130	3634	3020	2571	2715	2304	2547	5004	0001	5220	7100	0000	0174	4001	3640
016420	5216	0150	5761	3001	5067	7025	7715	7776	3102	1203	6242	7777	2023	3431	5113	2304	0126	1050	6100	0000
016424	0233	0000	0061	0004	6000	7160	0000	0151	6001	5050	0130	0164	2661	0004	6000	5110	0150	5203	0101	6425
016430	4361	4117	6103	1701	6704	2424	1764	1076	3402	0303	2045	7525	3006	2012	3244	5110	0150	5213	6125	4610
016434	2614	5552	5171	4000	7600	2023	7513	4016	7441	1642	2132	2637	6063	6301	3777	0266	0000	0061	0004	6000
016440	2113	0635	5063	4101	3666	6160	0000	0261	0004	6000	5614	0562	4136	6616	6446	3666	2074	5016	4425	4351
016444	0445	0164	4537	6624	6000	1343	6030	4016	4467	6710	5463	0547	5002	0001	6427	2112	2614	0000	0101	3666
016450	6251	7777	7463	6555	6661	5476	1666	1161	0004	6000	5577	6556	6655	7765	5666	5577	6675	5455	6665	5776
016454	5566	6074	5016	4524	6000	5566	1675	5106	5001	6455	5475	0020	0116	4274	6000	1001	1616	0777	7174	3761
016460	6130	0164	6104	0001	6675	6170	0000	7060	3300	0020	1560	7206	7253	1631	1656	2060	2634	6720	6022	2716
016464	3677	6632	7623	6251	1661	2234	1032	3016	7035	4610	1010	0666	0061	0004	6000	6130	0164	7004	0001	6675
016470	6033	0005	2071	0000	4210	4346	1156	1461	5077	7717	2016	0664	0061	7000	0070	0331	0165	0354	1312	0672
016474	2015	2737	1053	1633	7727	0460	0164	7603	2701	6703	1165	6206	0263	6672	0602	2271	6367	7663	2752	3625
016500	1166	1277	2236	6672	2261	5461	0032	2016	7032	2121	0400	0165	1061	0004	6000	2067	2531	6311	6562	0602
016504	6366	0206	0222	2163	6726	2326	0632	7523	6253	7762	1167	1666	6722	2615	4610	0322	0167	0322	1216	6441

15

016510	1561	4201	6003	3101	6503	5413	1762	4073	1103	6632	7326	0421	2143	7301	5667	2014	4366	6113	7775	4630
016514	5475	0040	0016	4274	6000	5627	1212	4454	4216	3420	2144	4733	3053	2107	3220	0313	0165	2053	3112	1322
016520	4307	1735	3036	1256	6211	3734	1032	3016	5234	3730	2076	0547	5004	0001	6427	6365	0541	2111	6026	5314
016524	3706	2673	6354	2116	3460	6160	0165	2702	0001	6753	1071	1730	0154	7715	4111	0310	0165	2674	3702	0330
016530	1263	5546	5004	0001	6427	1031	1203	4403	2301	6515	4376	1616	0000	0061	5417	2111	4643	0015	2172	0204
016534	5405	1223	6236	5322	2664	5637	1364	6421	3443	6054	0110	0000	0204	0001	6564	2111	4634	3054	4312	1444
016540	0311	0165	4154	1012	1122	5301	4632	1055	0016	3440	6150	0001	0007	1201	6550	4373	0761	2020	7601	2771
016544	5603	0547	5004	0001	6427	2121	4032	2016	5471	5327	2030	4132	2222	5633	6553	2266	2362	6254	0053	6052
016550	6622	5543	0006	2401	6557	0110	0001	0104	0001	6564	5420	0106	3321	2445	4600	1532	7234	6303	1401	6545
016554	5627	1745	0121	2443	6453	7311	0372	4220	2305	6030	1272	1547	5004	0001	6427	6742	4665	5110	6336	7554
016560	0115	0000	0004	0001	6564	5420	0662	0021	2445	4600	1532	7634	3007	4501	6554	0400	0165	4361	0004	6000
016564	6150	0165	4402	0001	6377	5415	0666	1043	4364	3560	2013	0151	1503	0101	6571	5021	0000	0115	5242	7605
016570	5120	0166	1340	6265	4611	7150	0000	5704	0001	6576	4343	6512	7000	0511	5624	2720	6513	0016	6134	0632
016574	6160	0000	0451	1700	0047	5362	0000	0071	5000	0057	5120	0166	1451	3001	6615	2052	2615	0000	0064	3052
016600	5411	1156	1471	7000	1750	0306	0166	1061	4077	7755	5443	1271	0637	7676	7205	0327	0166	0466	4005	4441
016604	4262	1401	2126	7012	0466	6622	5406	3673	6643	6464	0317	0166	0422	6241	5160	3671	5114	0620	4063	6647
016610	5461	0616	6777	7764	3436	0560	0166	0013	7775	4750	0400	0164	2761	0004	6000	1717	0110	6425	4742	3260
016614	1717	0631	4631	4631	4632	2000	0000	0000	0000	0012	5555	5555	5555	5555	5500	5555	5555	5533	3333	3300
016620	6130	0166	2104	0001	6675	5443	1736	4027	2022	4202	2760	6246	0644	6262	6676	2267	6730	6173	6403	7430
016624	7324	0422	2643	5306	3430	1544	5202	4436	6426	3200	4357	0074	2016	6554	6000	5063	0000	0076	0102	0057
016630	5023	0000	1771	4000	3720	6140	0166	5661	5077	7723	6170	0000	0413	7774	6000	6160	7777	1743	3614	6000
016634	0301	0166	4315	6132	0672	6336	1111	3620	1024	3260	6271	0000	7020	1025	4323	2261	1361	1663	3162	2673
016640	6371	0221	7315	6612	3232	0326	0167	0311	6236	4430	5063	0000	0061	7000	0004	5422	1734	4715	6253	6165
016644	0301	0166	4324	6712	3170	1262	1546	2073	1474	6000	0317	0166	4710	7114	6000	5634	0232	3112	6325	4630
016650	7617	5672	2122	2113	6612	5422	0206	0264	4206	3366	1562	5361	6561	7000	0004	0520	0166	4443	2445	4150
016654	2026	0113	2146	0003	6737	5475	0040	0016	4274	6000	0000	0000	0000	0000	0000	6150	0164	2761	0004	6000
016660	6130	0166	6102	0001	6676	2026	0106	2221	2602	1674	1346	2634	4037	1401	6407	6644	4603	3000	0204	3461
016664	1521	4202	7253	3236	6444	1114	2201	0271	2000	0010	6331	0234	3211	2436	3414	0302	0164	0771	2000	4000
016670	2314	2114	1315	6161	3764	0307	0166	7313	6317	6700	5463	0547	5002	5500	0000	7671	0030	6016	4072	0744
016674	5475	0025	5000	0004	6000	6150	0164	2761	0004	6000	4356	0201	3015	3152	0130	5243	0066	0072	6377	7677
016700	1534	5152	1520	1601	5115	2030	3523	3000	0001	1646	0326	0164	0702	3300	0000	6150	0164	2702	0001	6407
016704	2076	1242	3771	4000	0001	2047	2513	3016	7102	3234	5110	0150	5213	6125	4610	6363	0026	6000	0004	6000
016710	0000	0000	0000	0001	6745	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0400	0000	0261	0004	6000
016714	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0400	0000	0261	0004	6000	5110	0150	5153	5112	1160
016720	6120	0001	0054	4512	1444	6344	0201	0663	3102	0530	5110	0167	4310	0116	3550	0730	0167	3063	5546	7552
016724	6720	2560	5061	0004	6000	0120	0001	0004	0001	6733	5400	3665	5261	4477	7677	0110	0001	0004	0001	6732
016730	5605	0054	0016	7254	6000	0400	0167	6361	0004	6000	5500	3614	4000	1004	6000	6661	0073	0016	7356	7601
016734	5000	0000	7761	0004	6000	5410	0674	4110	6115	4603	5400	6054	0016	7357	6710	7160	0167	4551	6001	6710
016740	5110	0167	4410	6112	0714	5160	0060	3051	7000	6026	0200	0167	6361	0004	6000	0000	0000	0000	0000	1717
016744	0503	2355	1706	0657	0000	5110	0150	5153	5112	1160	6120	0000	0254	4512	1444	6344	0201	0663	3102	0530
016750	6160	0167	6306	3001	6752	5315	0532	5104	0001	6753	6355	4571	5157	2526	7202	5431	2544	2210	6112	2702
016754	5461	3547	2354	1325	4242	1063	3227	0454	6335	4743	5431	2544	2210	6112	2702	5461	3547	2361	4477	7767
016760	5413	2542	4210	6332	2704	5463	3547	4305	4001	6753	0266	0000	0061	0004	6000	7633	0366	5320	3062	0644
016764	5015	0000	1736	7135	4710	5465	0137	7751	7001	5051	0400	0164	2761	0004	6000	0000	0000	0000	0001	6000
016770	0000	0000	0000	0001	6006	0000	0000	0000	0001	6013	0000	0000	0000	0001	6014	0000	0000	0000	0001	6017
016774	0000	0000	0000	0001	6025	0000	0000	0000	0001	6033	0000	0000	0000	0001	6036	0000	0000	0000	0001	6041
017000	0000	0000	0000	0001	6045	0000	0000	0000	0001	6052	0000	0000	0000	0001	6065	0000	0000	0000	0001	6067
017004	0000	0000	0000	0001	6105	0000	0000	0000	0001	6120	0000	0000	0000	0001	6132	0000	0000	0164	6701	6414
017010	0000	0000	0000	0001	6136	0000	0000	0000	0001	5066	0000	0000	0000	0001	6240	0000	0000	0000	0001	6242
017014	0000	0000	0165	6501	6414	0000	0000	0000	0001	5066	0000	0000	0165	7201	6414	0000	0000	0000	0001	6270
017020	0000	0000	0000	0001	6274	0000	0000	0166	2001	6414	0000	0000	0000	0001	6302	0000	0000	0166	5701	6313
017024	0000	0000	0000	0001	6370	0000	0000	0000	0001	5066	0000	0000	0000	0001	6166	0000	0000	0164	5701	6414
017030	0000	0000	0000	0001	6315	0000	0000	0000	0001	6341	0000	0000	0000	0001	6276	0000	0000	0165	3101	6414
017034	0000	0000	0164	4701	6114	0000	0000	0164	4001	6414	0000	0000	0000	0001	6406	0000	0000	0000	0001	6406
017040	0000	0000	0000	0001	6406	0000	0000	0000	0001	6406	0000	0745	0300	0000	0000	0000	0000	0000	0000	0001
017044	0030	0000	0000	0001	7042	0000	0000	0000	0000	0000	0000	0000	0030	0000	0000	0000	0000	2605	0100	0000
017050	0000	0000	0000	0000	0000	0000	0000	0002	3577	7776	7777	7777	7777	7777	7760	2203	1400	0000	0000	0000
017054	0000	0000	0000	0000	0000	2401	4017	4000	0000	0540	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017060	2203	1400	0000	0000	0000	0000	0000	0000	0000	0155	0001	5070	0000	0000	0000	0000	0000	0000	0000	0001
017064	0030	0000	0000	0001	7062	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017070	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	5070	0000	0100	0000	0000	0000	0000	0000	0001
017104	0030	0000	0000	0001	7102	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

16

017110	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0171	2200	0001
017124	0030	0000	0171	2301	7122	0000	0000	0150	5300	0000	0000	0000	0000	6076	6135	0000	0000	0167	1200	0002
017130	0000	0000	0170	5000	5200	0000	0000	0170	4700	0000	0000	0000	0000	0001	7122	0000	0000	0000	0000	0002
017134	0000	0000	0000	0001	6711	0000	0000	2605	0100	0000	0030	0000	0000	0000	0000	0000	0000	0000	0000	0000
017140	0000	0000	0000	0000	0000	0000	0000	2605	0100	0000	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001
017144	0030	0000	0000	0001	7142	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0062	1100	0000
017150	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017160	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000
017164	0000	0000	0056	2000	0001	0030	0000	0150	5501	7163	0000	0000	0000	6000	0027	0000	0000	0000	0000	5200
017170	0000	0000	0062	2100	0003	0000	0000	0056	2001	6120	0000	0000	0150	5500	5600	0000	4000	0000	0000	0000
017174	0000	0000	0001	0000	0000	0000	0000	0001	0000	0000	0000	0000	0000	0000	5200	0001	0000	0000	2620	0160
017200	0041	0441	0031	0020	0011	0001	0000	0000	2620	0160	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000
017204	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001	0030	0000	0000	0001	7204	0000	0000	0000	0000	0000
017210	0000	0000	0000	0000	0000	0000	0000	0062	3100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017214	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017224	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0060	2000	0001	0030	0000	0030	0001	7225
017230	0000	0000	0000	4701	5067	0000	0000	0060	2577	6044	0000	0000	0062	4177	7776	0000	0000	0000	4701	6136
017234	0000	0000	0062	4100	6000	7777	7777	7777	7777	7776	0002	0000	0000	0000	3000	3000	0000	0000	0000	0000
017240	0000	0000	0000	0000	0001	0000	0000	0000	0000	0000	7777	0000	0000	0000	0000	0000	0000	0000	0000	6240
017244	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0000	4700	0001
017250	0030	0000	0062	2001	7246	0000	0000	0002	2001	6273	0000	0000	0150	6177	6053	0000	0000	0062	5100	0015
017254	0000	0000	0062	2001	6270	0000	0000	0062	5100	0200	0000	0000	0000	0000	0000	0000	6220	0000	0000	0000
017260	0001	0000	0000	0000	0000	2001	0000	0000	0350	0073	0000	0000	0000	0000	0200	0053	6700	6601	0000	0000
017264	3404	1401	6250	0044	1201	0000	6220	0000	0000	0000	0001	6427	0000	0000	0000	0001	6057	0000	0000	0000
017270	0000	0000	0000	0000	0001	0030	0000	0000	0001	7267	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017274	0000	0000	0062	6100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017300	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017304	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000
017310	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001	0030	0000	0000	0001	7310	0000	0000	0000	0000	0000
017314	0000	0000	0000	0000	0000	0000	0000	0062	7100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017320	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017330	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0012	2000	0001	0030	0000	2602	5601	7331
017334	0000	0000	0000	4701	5067	0000	0000	0012	2577	7774	0000	0000	0063	0177	7776	0000	0000	0000	4701	6166
017340	0000	0000	0063	0100	1200	0000	0000	0000	0000	0000	4000	0000	0000	2326	0256	3000	0000	0000	0000	0000
017344	0000	0000	0000	0000	0001	0000	0000	0000	0000	0000	7777	0000	0000	0000	0000	0000	0000	0000	0000	6300
017350	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0012	2000	0001
017354	0030	0000	0012	5001	7352	0000	0000	0170	3201	5066	0000	0000	0000	0077	6066	0000	0000	0063	1100	1224
017360	0000	0000	0012	5001	6276	0000	0000	0063	1100	1200	0000	0000	0000	0000	0000	0410	0000	0002	3260	2560
017364	0000	0000	0000	0000	0102	0000	0000	0000	0017	6750	0000	0000	0000	0000	7600	0066	0000	0101	0000	0000
017370	0000	0000	0000	0000	0102	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000
017374	0000	0000	0000	4700	0001	0030	0000	0062	4001	7373	0000	0000	0054	2001	6273	0000	0000	0060	2577	6053
017400	0000	0000	0063	2100	0015	0000	0000	0062	4001	6270	0000	0000	0063	2100	5400	0000	0000	0000	0000	0000
017404	0000	6240	0000	0000	0000	0001	0000	0000	0000	0000	0001	0000	0000	2617	0011	0000	0000	0000	0000	5400
017410	0053	1726	0000	0000	0030	0311	1726	0000	0000	0030	0000	6240	0000	0000	0000	0001	6427	0000	0000	0000
017414	0001	5070	0000	0000	0000	0000	0000	0006	2000	0001	0030	0000	0036	5001	7414	0000	0000	0170	3201	5066
017420	0000	0000	0000	0077	6066	0000	0000	0063	3100	0624	0000	0000	0006	5001	6276	0000	0000	0063	3100	0600
017424	0000	0000	0000	0000	0000	0410	0000	0001	5720	2560	0000	0000	0000	0000	0077	0000	0000	0000	0017	6750
017430	0000	0000	0000	0000	7600	0066	0000	0101	0000	0000	0000	0000	0000	0000	0077	0000	0000	0000	0000	0000
017434	0001	6427	0000	0000	0000	0001	5070	0000	0000	0000	0000	0000	0000	4700	0001	0030	0000	0063	2001	7435
017440	0000	0000	0060	2001	5067	0000	0000	0150	6177	7734	0000	0000	0063	4100	0016	0000	0000	0063	2001	6274
017444	0000	0000	0060	2000	0400	0000	0000	0000	0000	0000	0000	0000	0000	0000	6320	0001	0000	0000	0000	0000
017450	0002	0000	0000	0000	3000	0000	0000	0000	0000	6000	0054	7760	0000	0000	0000	3423	1230	0000	0000	0000
017454	0003	0000	0000	0000	3000	0001	6427	0000	0000	0000	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001
017460	0030	0000	0000	0001	7456	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0063	5100	0000
017464	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017474	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0000	0002	0000	0000	0000
017500	0000	0000	0000	6000	0001	0030	0000	0000	0001	7477	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017504	0000	0000	0063	6100	0000	0000	3000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

97404700A

017510	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017514	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000
017520	0001	5070	0000	0000	0000	0000	0000	0060	2000	0001	0030	0000	0030	0001	7520	0000	0000	0000	4701	6273
017524	0000	0000	0060	2577	6044	0000	0000	0063	7177	7776	0000	0000	0000	4701	6136	0000	0000	0063	7100	6000
017530	6000	7777	7777	7777	7777	0002	0000	0000	0000	3000	3000	0000	0000	0000	0000	0000	0000	0000	0000	0001
017534	0000	0000	0000	0000	0000	7777	0000	0000	0000	0000	0000	0000	0000	0000	6370	0000	0000	0000	0000	0000
017540	0001	6427	0000	0000	0000	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001	0030	0000	0000	0001	7541
017544	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0064	0100	0000	0000	0000	0000	0000	0000
017550	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017560	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001
017564	0030	0000	0000	0001	7562	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0064	1100	0000
017570	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017600	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000	0001	6057	0000	0000	0000
017604	0000	0000	0000	0000	0001	0030	0000	0000	0001	7603	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017610	0000	0000	0064	2100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017614	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017620	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0001	6427	0000	0000	0000
017624	0001	6057	0000	0000	0000	0000	0000	0000	0000	0001	0030	0000	0000	0001	7624	0000	0000	0000	0000	0000
017630	0000	0000	0000	0000	0000	0000	0000	0064	3100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017634	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
017644	0001	6427	0000	0000	0000	5555	0000	0000	0000	0001	0311	1700	1073	0000	0312	0100	2217	0200	0533	3045
017650	1202	0536	3024	1214	0430	1075	3401	3044	2200	6000	5301	1254	3444	3045	0416	0321	5000	1201	2200	7477
017654	2300	0300	5400	1201	1402	3424	3044	1015	0635	1404	3524	0332	1404	3424	3045	1377	0403	1007	0623	1400
017660	3410	3411	3412	3060	1237	3413	3061	3414	3053	1006	3155	1006	3154	1602	6210	1601	6210	3624	1277	2300
017664	0000	3424	3057	0402	6220	1401	3544	3052	0505	3044	2200	1777	3444	3053	1006	3155	1006	3154	6240	2000
017670	0000	0506	1444	0200	0364	0314	1400	3411	3025	5400	1254	3077	6370	1250	1460	0200	0364	0100	0103	0000
017674	0000	0000	0000	0000	0020	0030	1030	0000	0000	5400	1270	2035	0311	0200	0533	0000	2000	0000	3444	1074
017700	1207	1107	0412	1102	0504	6030	0100	1507	0100	1145	0100	1456	3022	0474	3074	1621	6030	3057	1701	6001
017704	3005	1337	3434	1071	1102	0434	1101	0556	2000	1377	3406	1404	3506	4006	0447	3301	0572	5006	0001	3302
017710	0566	5006	0002	3303	0562	5006	0003	3304	1377	0555	5006	0003	1277	3423	1400	3422	3033	1277	1014	1102
017714	0200	1421	3010	3424	3057	6220	1701	6230	0100	1473	2025	1603	1000	0000	2025	1603	1002	0001	2043	0000
017720	0000	0002	0000	0100	1420	3400	1066	3500	1404	6010	3013	1014	3114	3100	6010	0363	5000	1261	1006	0612
017724	1400	6010	3053	1006	3155	1006	3154	1606	6210	1414	3424	0100	1104	3045	2300	1000	0503	0200	3213	2000
017730	3062	3415	2033	0406	0200	0533	1400	3457	0100	1145	3045	2300	6400	0405	2035	0306	0100	1102	3022	0405
017734	3021	3422	1401	3423	5000	1261	1006	0615	1400	3430	3431	3071	3432	3053	1006	3155	1006	3154	1606	6230
017740	0100	1145	5000	1261	1006	0624	3053	1006	3155	1006	3154	1606	6010	3025	1006	3512	1063	3511	3053	1006
017744	3155	1006	3154	1606	6210	1452	5400	1225	3045	1202	0503	0100	1104	2000	0000	1112	1601	0506	3020	1277
017750	3405	0200	1613	0100	1125	0100	1612	5000	0551	3105	6010	3401	3010	2200	7577	2300	0200	3410	3001	6210
017754	0360	0100	1632	0200	0547	3027	5400	0576	0370	5000	1261	1011	0706	0100	1646	0671	1406	0302	1434	3401
017760	5000	0575	3410	3077	6204	6170	1705	3076	6010	3010	0405	1477	1701	0576	0370	3001	1134	0404	3004	0200
017764	0446	3001	0100	1262	0000	0000	0040	4204	0044	4300	0050	6400	0054	6400	0060	6400	0070	5700	0200	0000
017770	0204	0002	4210	0000	4214	0002	4224	0002	4230	0000	4234	0002	0240	0104	0250	0000	2260	0000	2264	0002
017774	0600	0000	0640	4204	0000	1523	0302	2424	0307	1524	0310	1624	0310	0322	2203	0000	2100	1523	0304	2424
020000	0307	1524	0310	1624	0310	1420	1420	1421	1420	0320	2003	0000	2133	5523	1477	1524	0310	1624	0310	0000
020004	2066	1400	3457	3420	3045	1377	0403	0100	1145	0200	2160	0504	1412	0100	1262	3044	3430	2000	3062	3415
020010	2033	0206	0200	0533	0306	1400	3457	1414	0100	1262	3030	3444	3057	1701	6010	0100	2464	3045	2300	6400
020014	0503	0100	1507	1400	3424	0100	1104	0100	1435	1401	3447	1063	3446	0100	2106	3046	1014	3347	0466	2377
020020	7777	0567	3045	1006	0605	2000	6400	3445	0360	3045	1314	1114	3445	0347	3053	1006	3155	1006	3154	1602
020024	6010	1601	6210	3044	5300	1260	1277	1150	0503	1410	3544	0100	1104	1400	0100	2157	1440	3401	4001	1071
020030	0420	1745	0666	4001	1277	0415	1745	0661	3601	1143	0563	3043	1071	1745	0753	0351	4001	0547	3601	1143
020034	0573	3043	1071	0541	3040	0340	1400	3457	3435	3425	1430	3426	2000	4007	3427	3053	1014	3447	1063	1237
020040	3453	1006	3155	1006	3154	6040	1601	6030	1603	6063	3052	1006	3347	3447	1063	3446	3033	1014	1277	3452
020044	3033	1237	3460	3034	3461	3053	1014	3354	1604	3152	1071	3256	0704	1400	0100	1262	2000	1705	3425	3044
020050	1204	1076	3445	3044	1740	0734	1402	3525	4025	0505	2035	0301	0200	0533	3344	2200	0774	0564	5025	0001
020054	3445	1377	0413	3043	1274	1174	0505	3045	1314	1110	3445	0200	2107	3044	1202	5500	1206	3043	5400	1260
020060	1377	3443	3044	2200	0774	3444	1404	6020	3063	3220	0713	3063	3221	0610	3063	6010	1601	3457	3020	3422
020064	0310	1402	3520	6010	3321	0503	0100	2022	3010	0467	3340	0565	3014	3351	1237	0561	3011	3341	0556	3012
020070	3342	0553	3013	3343	1377	0547	3020	3322	0403	3620	3457	3077	6210	3057	3414	1400	3413	1456	0200	0364
020074	3011	0406	1400	3457	1414	0100	1262	3057	6020	1701	6010	3024	2200	7700	2300	0000	5400	1201	3013	1204
020100	0411	3045	1071	1257	1157	0404	1410	0100	1262	3020	1277	5100	0551	6001	3004	2200	3777	3430	3001	1006
020104	0607	4025	1007	0624	1430	0100	1262	3030	2300	2424	0403	2000	4000	3530	5400	3004	3031	2200	3777	3431

18

020110	4025	1006	0755	3045	1202	0417	2000	1772	5400	3003	3014	1071	5400	1577	3013	1201	0425	1424	0100	1262
020114	3045	1377	0417	1071	1257	1157	0405	3045	2300	1000	0503	0100	1456	2000	2012	5400	3000	3052	0503	3100
020120	2712	3030	2300	5524	0404	2300	0300	0527	3022	1067	1101	3407	1112	0404	1107	0517	1420	1640	2100	4002
020124	3430	3022	2200	0200	1076	3530	3024	1064	1002	3530	3057	1701	3463	3053	1006	3155	1006	3154	1601	6230
020130	1603	6263	3031	5400	1261	1076	0617	3052	1702	0714	3053	1006	3155	1066	3154	1606	6030	3033	3134	0402
020134	3635	3053	1006	3155	1006	3154	1602	6001	1601	6010	3014	3465	3066	1237	3466	1014	3367	1701	1071	3256
020140	0637	3013	1237	3464	3266	1014	3165	3267	0627	3064	3260	1014	3165	3261	0721	3005	3463	3004	1237	3462
020144	3266	1014	3163	3267	0607	3062	3260	1014	3163	3261	0604	1404	0100	1262	2000	1756	3401	4001	2300	1523
020150	2200	3777	0406	1402	3501	4001	0567	0326	5000	3004	2300	1523	0516	5000	1261	1011	0612	3052	1702	0607
020154	5000	1261	2200	7377	5400	1261	4001	1006	0606	5001	0001	3401	0101	0000	5001	0001	2335	0000	0100	1102
020160	3503	0100	3055	0000	0047	0100	3062	2000	3147	3401	4001	3344	2200	0774	0410	1402	3501	4001	0570	1422
020164	0100	1262	2000	1705	3425	4001	4425	5001	0001	5425	0001	2200	3777	2300	0500	0516	5400	2030	5400	2031
020170	2000	7102	5400	2466	2000	0400	5400	2470	0310	2300	0300	0505	2000	7100	5400	2466	0100	2327	0100	0500
020174	0104	0500	2110	0500	0114	1000	0120	0500	0130	0600	0140	4500	0144	4500	0150	4600	0160	4500	0170	4600
020200	0174	4600	0300	0500	0330	0600	0340	4500	0350	4600	0370	4600	0000	0100	3212	5000	3004	2300	1523	0571
020204	3057	1701	6010	3013	1201	0563	3014	1071	3401	1717	0656	5001	3337	3453	5400	3246	3022	0426	3021	0424
020210	0100	3245	1400	3407	3423	3422	3424	3020	1277	3411	3021	2200	3777	3412	3007	3413	1443	0200	0364	0100
020214	1145	3406	3422	3020	1277	3405	1401	3423	3407	0200	1633	3004	0200	0437	0200	0606	2000	3335	0200	0612
020220	0603	0200	1647	3004	0200	0446	5000	1201	2200	7477	2300	0300	5400	1201	0100	3254	0000	0000	3247	0000
020224	3247	3247	0000	0000	3247	3247	0000	0000	3272	0000	0000	0000	0000	0000	0000	3503	0200	1705	0000	0121
020230	0100	1712	3022	0503	0100	1435	3035	0405	2035	0306	0200	0533	0200	2342	1071	0506	3044	1074	1102	0554
020234	3424	1400	3436	3425	3044	1074	1277	1101	0405	2035	0303	0200	0533	3020	1277	3405	3022	3406	3023	3407
020240	1400	3430	3431	3432	3062	3433	3063	3434	0200	2362	1404	6010	3610	6010	3010	3320	1277	1014	3311	3321
020244	0512	3411	1422	0200	0364	3012	3405	0200	1633	0306	0200	1633	3004	0200	0437	0200	0606	0100	2111	1477
020250	1701	0576	3076	6010	3010	0571	3004	0200	0446	3006	3422	3007	3423	3024	1214	3236	0603	1400	3424	0100
020254	1537	1414	3424	5000	6767	3401	0422	3407	1006	0605	1071	3406	1400	3407	1410	3424	0100	2107	3001	1006
020260	0607	0200	0606	0304	3725	0100	2034	3064	3233	1014	3165	3234	1701	0603	2100	0000	2177	7677	0762	1071
020264	3403	3625	3256	0654	2000	6766	0200	0616	0603	0200	1647	5000	6766	0503	0100	2057	3407	1006	0607	1071
020270	3406	1400	3407	0200	0606	3055	2300	0000	0403	0200	2362	5000	6767	3401	0527	0100	2335	1013	0712	1011
020274	3101	3402	5400	6767	1400	5402	6770	0306	2000	0500	5400	6767	1400	5400	6766	1402	3501	3134	1006	3133
020300	1014	2177	7777	0621	3201	2100	0000	6301	6770	2177	7777	3434	1063	3433	3003	1102	0503	3076	6226	0336
020304	3402	3001	3202	3403	1002	3103	2100	6770	5400	2274	3033	1014	3134	2100	0000	6303	6770	3002	0406	2000
020310	0000	6302	6770	3002	2100	0000	3434	1063	3433	2000	0000	6230	3013	0517	2000	0000	6010	3013	1237	3464
020314	1014	3314	3465	2177	7777	0703	0100	2107	3001	2300	0100	0504	3424	0100	2111	1404	3424	0100	2107	0100
020320	2341	3064	3262	1014	3165	3263	1701	0667	1006	3166	3260	1014	3167	3261	0360	0100	2361	3066	3260	1014
020324	3167	3261	5400	2121	1063	2300	2100	5400	2120	2000	2177	3366	5400	2223	5400	2321	1500	3367	5400	2224
020330	5400	2322	3061	5400	2277	3060	2300	2100	5400	2276	3055	5400	2161	1006	5400	2264	2300	7777	5400	2234
020334	1063	2300	2100	5400	2263	1177	5400	2233	3060	1006	3155	1006	3161	5400	2272	1063	2300	2000	5400	2271
020340	3053	1006	3155	1006	3154	1602	5400	2304	5400	2311	1063	2300	2000	5400	2303	5400	2310	5600	2311	1063
020344	5500	2310	3066	1006	3155	1006	3167	5400	2230	1063	2300	2100	5400	2227	0100	2361	0000	0000	0000	0000
020350	3503	0300	2526	0000	0150	0100	2533	2000	2556	3401	1402	3501	4001	0505	1414	3424	0100	1537	3344	1303
020354	0565	5001	0001	3401	0101	0000	0001	2577	0020	2623	0200	2761	0210	3064	0230	3060	0250	2614	0600	2605
020360	0000	2000	2107	5400	2334	0100	2533	1400	5400	2076	5400	2077	1414	0302	1410	3436	2000	2111	5400	2340
020364	0360	2000	2631	5400	2110	0100	2533	3024	1214	0407	1210	0503	0100	2736	0100	2034	2000	2705	5400	2334
020370	5400	2340	2000	2034	5400	2110	1404	3424	3436	2000	6766	0200	3473	5400	2706	0420	3033	3462	3034	3463
020374	0200	2342	3402	3201	0604	3002	3401	0404	3001	0100	2217	2000	0000	1071	0426	5600	2760	1237	0513	3004
020400	0200	0446	0200	0424	3004	0200	0437	0200	0606	2000	6766	0200	3473	5400	2706	0347	5000	6766	0407	5000
020404	1260	1274	1174	0511	0342	1410	3424	5000	6767	0503	1414	3424	0100	2034	0000	2000	6763	5400	2232	5400
020410	2255	5400	2266	1502	5500	2123	1402	5500	2330	2000	0303	5400	2170	1400	5400	6763	5400	6765	5400	7470
020414	3071	5400	6764	5000	1260	1274	3543	1277	1174	0407	2000	3034	5400	2077	0100	2612	0100	2614	3001	1006
020420	0603	0200	0606	3055	5300	2161	0403	0200	2362	1417	5400	6770	1400	5400	6767	3401	0100	2213	2000	3213
020424	5400	2334	2000	3213	5400	2340	2000	3430	5403	2056	2000	3533	3437	3053	1006	3155	1006	3154	1605	6010
020430	3013	1237	3434	1014	3314	3435	1071	3256	0704	1426	0100	1262	3013	1006	0613	3022	4437	3023	5437	0001
020434	1402	3537	3635	1063	3534	3020	1277	3405	0200	1633	3014	1003	1620	2177	6777	5400	3253	5400	3366	5400
020440	3405	1163	5500	3252	1277	5500	3365	1277	5500	3404	0200	3240	1400	4437	3037	2300	3533	0412	3337	3437
020444	4037	3422	5037	0001	3423	0100	2533	0100	3430	1402	3537	4037	0505	1410	3424	0100	2034	5037	0001	3407
020450	4037	3306	3405	3306	3406	0200	0606	0100	2111	0100	3237	1401	3433	1063	3432	1401	3407	3021	3406	1075
020454	2100	7777	6010	3006	1203	3401	5001	0010	3403	1006	0610	3006	1074	1201	3401	5001	0601	3403	3037	2177
020460	1177	0703	0100	3237	3034	1006	3155	1006	3135	6010	3014	3431	3113	0464	3013	1237	3430	3030	3232	1014
020464	3131	3233	0603	0100	3241	3107	5437	0001	3203	0621	3006	4437	3030	3432	3031	3433	5037	0001	3407	3635
020470	1063	3534	1402	3537	0100	3274	3003	3207	3533	1063	3532	1400	3407	3066	1075	2100	7777	6010	3006	1203
020474	3401	5001	0010	3406	1006	0704	1426	0100	1262	1067	2100	7777	6010	3006	1203	3401	5001	0010	3403	1006
020500	0610	3006	1074	1201	3401	5001	0601	3403	0100	3317	3053	1006	3155	1006	3154	1605	6010	3037	2177	4244
020504	1076	3514	1063	3113	1237	3413	4037	3405	2000	4000	3513	1410	1110	3424	3053	1006	3155	1006	3154	1605

27-20

020510	6210	0100	1537	3001	0100	3472	3403	0200	0616	0603	0200	1647	5003	0001	3401	4003	3505	3401	5003	0001
020514	0457	3407	1006	0653	1071	3406	1400	3407	0200	0606	4003	0403	5003	0001	3401	0100	3472	0000	0000	0000
020520	3503	0410	1705	0000	0247	0100	1712	3044	5300	1260	1277	1150	0503	1410	3544	3062	3264	1014	3163	3265
020524	0607	1006	3166	3260	1014	3167	3261	3416	1071	3402	0523	3044	1074	0443	1207	1101	0440	1103	0413	3016
020530	0511	3024	1206	1102	0405	5400	2135	5400	2136	3056	3446	3044	1104	0507	5400	2430	5400	2431	1401	3446
020534	1220	0416	3002	3446	1400	5400	2210	5400	2211	5400	2424	5000	3337	5400	2403	3044	1012	0610	1010	1277
020540	0405	3024	1302	1102	3424	3044	2177	7573	0710	0403	1014	0605	2035	0305	0200	0533	3035	0405	2035	0306
020544	0200	0533	3023	3407	3044	1074	1203	0502	1601	3107	3546	0200	2435	1400	3045	1377	0404	1400	5400	2402
020550	1400	3425	3430	3431	3432	3064	3433	3065	3434	0200	2772	3004	0200	0437	0200	0606	3006	1074	1201	3401
020554	5001	0601	3437	0311	1400	5400	0576	0200	3342	3027	5400	0576	0200	2150	0200	3224	3006	3422	3007	3423
020560	0100	1537	0100	2147	3071	5400	6767	1600	3435	0314	2000	2163	5400	2431	5000	7501	2177	7277	0403	0100
020564	4064	3062	3233	1014	3163	3234	0603	2100	0000	3401	3235	1014	3403	1006	0620	0100	2147	3001	5400	6767
020570	3435	0511	0100	2327	1702	0605	5000	3340	5400	2163	3033	1014	3134	3135	2177	7777	0623	2100	0000	3235
020574	6135	6770	2177	7777	3434	1063	3433	3003	1277	1102	0503	3076	6226	0100	2324	3402	3035	3202	3401	0410
020600	3033	1014	3134	2100	0000	6101	6770	3002	0505	0320	2000	0000	0317	3001	1002	3101	2100	6770	5400	2314
020604	2000	0000	6102	6770	3002	2100	0000	3434	1063	3433	3603	0303	0100	4006	3007	1601	5400	6766	3337	0505
020610	3636	4036	5400	6766	2000	6766	0200	0612	0603	0200	1647	5000	6766	3407	1006	0616	1071	3406	1400	3407
020614	0200	0606	3006	1074	1201	3401	5001	0601	3437	3055	2300	0000	0403	0200	2772	2000	0000	6230	3003	1277
020620	0515	2000	0000	6010	3013	1237	3462	1014	3314	3463	2177	7777	0611	3625	3256	0606	3035	1071	0403	0100
020624	2172	0100	2147	0100	2434	3057	5400	2665	1701	6010	3013	1230	3416	3022	0523	5400	2124	3646	3021	0516
020630	5400	2524	3046	3413	1402	0200	0364	3020	1377	3311	3420	3014	3421	3406	3020	1277	3405	0200	1633	3014
020634	1003	1620	2177	6777	5400	2542	5400	2652	1063	2300	2100	5400	2541	5400	2651	2000	3406	3436	3006	0312
020640	1400	5400	3240	3036	5400	3243	3006	0100	2644	3402	4436	1075	2100	7777	6010	3002	1074	1201	3401	3002
020644	1203	3433	5030	0010	3403	1006	0615	3046	5201	0601	0705	3446	3636	3003	0346	0200	2756	0100	2664	3036
020650	5400	3243	3003	5400	3240	3246	0664	3016	0416	1015	0704	1416	0100	1262	3002	3306	0504	3003	3323	0403
020654	0200	2756	3046	5201	0601	3446	0735	0434	3005	3411	3636	3002	3412	3046	3413	1455	0200	0364	3014	0434
020660	1006	1071	4436	1075	2100	7777	6010	4036	1203	3401	3636	5001	0010	1006	0762	2000	0001	0505	3410	3074
020664	1635	6210	2000	3406	3436	0100	2434	3005	1003	3305	1370	2100	3333	5400	2744	3074	1635	6373	2743	0200
020670	0424	3011	0516	5400	2665	5000	3243	3436	3021	0405	0200	1647	0100	2631	0100	2457	2000	2743	0200	0501
020674	0100	1712	0521	3030	5524	2201	0313	5514	1115	1124	5700	0000	0100	2755	3016	0474	1016	0604	3024	1202
020700	0567	1420	0100	1262	0100	2771	3066	3260	1014	3167	3261	5400	2201	1063	2300	2100	5400	2200	2000	2177
020704	3366	5400	2233	5400	2417	1500	3367	5400	2234	5400	2420	3066	1014	3167	1701	5400	2300	1063	2300	2000
020710	5400	2277	3061	5400	2317	3060	2300	2100	5400	2316	3055	5400	2374	1006	1600	5400	2271	2300	7777	5400
020714	2244	1063	2300	2100	5400	2270	1177	5400	2243	3060	1006	3155	1006	3161	5400	2312	1063	2300	2000	5400
020720	2311	3053	1006	3155	1006	3154	1602	5400	2401	5400	2407	1063	2300	2000	5400	2400	5400	2406	5600	2401
020724	1063	5500	2400	3066	1006	3155	1006	3167	5400	2237	1063	2300	2100	5400	2236	0306	5600	2234	1063	5500
020730	2233	0100	2771	0100	3160	1400	5400	6766	3007	1601	5400	6767	3337	0505	3636	4036	5400	6767	2000	6766
020734	0200	0612	0603	0200	1647	3625	5000	6767	3407	1006	0644	1071	3406	1400	3407	0200	0606	0370	0100	3223
020740	3044	1074	1207	1103	0503	0200	3161	3024	1202	0414	2000	7777	3402	2000	7777	3401	3236	0731	0504	3007
020744	3202	0625	3076	6010	3010	0405	1477	1701	0576	0370	3004	0200	0446	3001	2300	7777	0405	3002	3413	4001
020750	0326	0337	1430	0200	0612	0603	0200	1647	3076	6010	3010	0405	1477	1701	0576	0370	3004	0200	0446	3007
020754	3413	3006	2200	3777	3412	3005	3411	1443	0200	0364	5700	1601	0100	3223	0317	0363	0100	3341	2000	3777
020760	5400	6766	1477	5400	6767	3057	0406	5400	7001	1701	6170	6770	3005	5400	6775	3006	5400	6776	1427	6170
020764	7002	1400	3407	0200	0606	2000	6766	0200	0612	1401	3407	0100	3341	0000	0000	3503	0500	3706	0000	0034
020770	0100	3713	2000	3734	3401	1402	3501	4001	0503	0100	1537	3344	1303	0567	5001	0001	3401	0101	0000	0204
020774	3741	0000	1400	5400	2156	5400	2276	5400	3150	5600	3053	2000	0612	5400	2207	2000	0303	5400	2425	1402
021000	5500	2154	2000	3702	5400	2315	3016	0503	0100	1712	3064	1006	3155	1006	3165	6170	7475	0100	3713	2000
021004	0303	5400	2324	2000	2151	5400	2431	5000	6767	3402	1002	3502	5002	7001	5400	7501	5002	6770	1117	0403
021010	0100	2327	3002	0422	2000	4047	5400	2431	5400	2434	0100	2327	2000	2151	5400	2431	2000	2147	5400	2433
021014	3725	0200	3161	0100	2363	1701	0706	0200	3224	1402	0100	1262	2100	0501	3402	1001	3102	3403	1003	3102
021020	1002	3103	1066	5400	6767	1602	3435	1400	5400	2324	0100	2172	0000	0000	0000	3503	0600	4117	0000	0266
021024	0100	4124	0200	5651	0100	4210	3006	3422	3007	3423	5000	1261	1006	0662	3052	1702	0757	3045	1377	0404
021030	3021	0200	5603	5000	1270	0407	3053	1006	3155	1006	3154	1606	6030	3037	1006	3432	1063	3431	3036	1006
021034	3531	1063	3430	3053	1006	3155	1006	3154	1606	6230	0100	4124	3033	3436	3034	3437	3022	0406	3021	3406
021040	0200	5351	0604	1426	0100	1262	3045	1202	0405	3024	1302	1102	3424	3033	3436	3034	3437	0100	4132	3032
021044	1240	0542	3044	1012	0604	1010	1270	0571	3033	1237	3433	1014	3334	1702	0705	1602	1071	3256	0704	1432
021050	0100	1262	3022	0505	3402	1401	3403	0305	3021	3406	0200	5467	3033	1006	3155	1006	3134	6010	3002	3413
021054	3436	3003	3414	3437	3033	1006	3155	1006	3134	6210	0100	4132	3447	1063	3446	3006	3321	0504	3007	1101
021060	0421	3707	0606	0200	5012	3001	1701	3407	3046	1014	3147	1701	0553	1400	3424	0100	4132	1450	3444	0371
021064	3447	1063	3446	7006	3321	0504	3007	1101	0464	0200	4632	3045	1210	0404	3001	1006	0662	3046	1014	3147
021070	1701	0552	3424	0100	4132	3045	1214	3424	1114	0505	0200	5467	0100	4132	5000	0601	5400	4503	3004	0200
021074	0437	0200	0606	2000	6766	0200	5426	1071	0524	5000	6766	5100	6767	0422	3045	1210	0404	5000	6766	0511
021100	3046	1014	3147	1701	0407	3447	1063	3446	5700	4503	0544	3004	0200	0446	0200	0424	3011	0506	2000	4502
021104	0503	0100	4426	1404	3424	5000	6766	0510	1414	3424	5000	6767	0403	1410	3424	0100	4132	3044	5400	1272

97404700A

40

9740470OA

033164	7762	3002	1207	1036	5500	7763	3096	1292	1076	3500	1590	3401	3601	1530	3500	5674	1630	1277	1017	5592
033170	7766	1066	1377	5500	7767	3016	1021	3603	1411	3501	3001	1074	1207	5500	7764	3001	1237	1036	5500	7765
033174	0100	7522	1017	2532	5503	3333	3355	2433	3355	2333	3357	2000	0000	0000	0000	4223	1100	7503	0000	0045
033200	0102	7512	3004	5500	7525	5400	7636	1277	5500	7527	1277	5500	7531	7700	0013	7400	1414	7100	7512	0200
033204	7702	3077	6370	7670	6370	7672	1702	6170	7714	6170	7707	3005	1003	3305	1370	5500	7671	5400	7715	2000
033210	7673	3402	2000	7512	3410	4010	1071	0200	7656	4010	1277	0200	7656	3610	2177	0257	0764	5400	7733	1410
033214	3502	1506	5500	7577	2310	0243	0552	3001	1011	0631	5000	0551	3105	6010	3114	1003	1602	6204	3001	1013
033220	5706	1031	0614	1400	0100	0615	7700	0114	5000	0702	1604	5400	7654	0306	3004	0200	0446	2006	7670	0190
033224	7510	0100	7655	3400	1003	3300	1370	2100	3333	4402	3602	0365	0411	3333	5655	5555	5555	5555	5555	1400
033230	3401	0100	7701	5000	0573	3401	1014	0671	5000	0633	1701	3400	2177	0775	0660	5000	7525	3402	2177	0375
033234	0654	3600	1514	3502	0674	1513	3202	1277	5500	7747	1614	1277	5500	7764	3002	5500	7755	5000	7521	1000
033240	4300	4400	3600	2000	7777	1076	5400	7763	5000	7521	2200	0000	1000	4300	4400	0100	7701	0000	0000	0000
033244	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0104	0300	1073	4201	0001	0201	2400	1073	4201	0013
033250	0310	0400	1073	4201	0025	0315	2300	1073	4201	0035	0320	1500	1073	4201	0041	0323	3400	1116	4201	0045
033254	0411	2300	1073	4201	0053	0417	0700	1073	4201	0066	0423	3400	1073	4201	0072	0530	2500	1073	4201	0101
033260	1115	2300	1073	4201	0103	1725	2400	1073	4201	0110	1735	4100	1073	4201	0111	2006	2500	1073	4201	0124
033264	2111	2300	1574	4201	0134	2135	4100	1574	4201	0146	2306	1500	1073	4202	0005	2314	1400	1073	4202	0014
033270	2315	2000	1073	4202	0024	2414	3000	1073	4202	0026	2722	1500	1073	4202	0027	3023	2000	1073	4202	0034
033274	3322	1600	0000	4202	0036	3403	0400	1274	4202	0037	3403	1300	1073	4202	0050	3404	2300	1073	4202	0053
033300	3405	0400	1073	4202	0062	3414	2400	1073	4202	0073	3422	2000	1073	4202	0074	3424	0400	4700	4202	0075
033304	3424	2000	1073	4202	0102	3424	2300	1073	4202	0103	3503	1100	5650	4202	0107	3503	1200	5650	4202	0111
033310	3514	2000	1773	4202	0112	3520	0300	1773	4202	0115	3522	0300	1773	4202	0120	3523	0100	1134	4202	0124
033314	3523	0200	1134	4202	0126	3523	0300	1134	4202	0136	3523	0400	1134	4202	0137	3523	0500	1134	4202	0140
033320	3523	0600	1134	4202	0142	3523	0700	1134	4202	0143	3523	1000	1134	4202	0144	3524	0400	1073	4202	0145
033324	3524	1200	1773	4203	0004	3601	0100	1351	4203	0006	3601	0200	1351	4203	0011	3601	0300	5053	4203	0014
033330	3601	0400	1372	4203	0015	3601	0600	5053	4203	0017	3602	0100	1773	4203	0021	3602	0200	1773	4203	0022
033334	3602	0300	1773	4203	0026	3611	0100	2077	4203	0030	3611	0200	3207	4203	0032	3614	0100	1527	4203	0035
033340	3614	0200	1527	4203	0036	3614	0300	1527	4203	0037	3614	0400	1527	4203	0042	3614	0500	1527	4203	0044
033344	3614	0600	1527	4203	0046	3615	0100	2372	4203	0050	3615	0200	6576	4203	0054	3615	0300	2372	4203	0056
033350	3615	0500	2154	4203	0063	3615	0700	2653	4203	0066	3615	1000	7274	4203	0067	3615	1100	3042	4203	0070
033354	3615	1200	2404	4203	0075	3615	1300	3772	4203	0077	3615	1500	2651	4203	0100	3615	1600	7274	4203	0101
033360	3615	1700	3045	4203	0102	3615	2000	3775	4203	0105	3620	0300	2041	4203	0106	3620	0500	1365	4203	0107
033364	3620	0600	4700	4203	0113	3620	1000	1365	4203	0115	3620	1200	4700	4203	0120	3622	0700	5070	4203	0121
033370	3623	0100	3061	4203	0122	3624	0100	2564	4203	0123	3624	0200	2564	4203	0124	3624	0300	2564	4203	0125
033374	3624	0400	2564	4203	0127	3624	0500	2564	4203	0131	3624	0600	2564	4203	0132	3624	0700	2564	4203	0134
033400	3624	1000	2564	4203	0135	3624	1300	2564	4203	0140	3624	1400	2564	4203	0141	3624	1500	2564	4203	0142
033404	3625	0100	3650	4203	0143	3704	0100	4773	4203	0147	3704	0200	5000	4203	0151	3704	0300	4223	4204	0001
033410	4401	0100	0000	4204	0002	4402	0100	6173	4204	0043	4405	0100	6611	4204	0063	4407	0100	6174	4204	0114
033414	4411	0100	1773	4204	0123	4412	0100	7773	4204	0143	4422	0100	6756	4204	0147	4424	0100	6544	4205	0044
033420	4432	0100	3100	4205	0572	0300	0001	3150	0047	1073	0000	0000	0000	0000	0000	2331	2305	0411	2400	0001
033424	0556	0000	0000	4004	0354	4004	0000	0000	0000	0231	0310	0523	2300	0000	0000	0000	0000	0000	4006	0117
033430	0120	2005	1604	0000	0012	0012	0000	0000	4011	0014	0124	2401	0310	0000	0000	0310	0116	0705	0000	0000
033434	0405	2611	1605	0000	0000	0705	2400	0000	0000	0000	2001	0313	1601	1500	0000	2005	2215	1124	0000	0000
033440	2025	2207	1500	0000	0000	2205	2014	0103	0500	0000	2301	2615	0000	0000	0000	4010	0000	0000	0000	0000
033444	2006	0124	0334	0000	0000	0000	0000	0000	4011	0025	2006	0301	2434	0000	0000	0000	0000	0000	4011	0046
033450	2006	0317	2031	3400	0000	0010	0000	0000	4011	0105	2006	1425	1520	3400	0000	0500	0000	0000	4011	0127
033454	2006	1417	0104	3400	0000	0000	0000	0000	4012	0023	2006	2300	0000	0000	0006	0025	0000	0000	4012	0113
033460	2006	1417	0104	0000	0000	2006	0425	1520	0000	0000	2006	0301	2400	0000	0000	2006	0124	0300	0000	0000
033464	2006	0317	2031	0000	0000	4004	0000	0000	0000	0546	2025	2207	0114	1400	0000	0015	0000	0000	4012	0137
033470	1523	1100	0000	0000	0002	0053	0000	0000	4013	0004	0614	0127	0000	0000	0000	4004	0000	0000	0000	3230
033474	0103	0317	2516	2400	0002	0016	0000	0000	4014	0032	0601	1511	1431	0000	0000	4016	0000	0000	0000	0260
033500	0310	0122	0705	0000	0001	0011	0000	0000	4014	0037	4056	0000	0000	0000	0561	1123	0600	0000	0000	0001
033504	0017	0000	0000	4014	0050	4004	0000	0000	0000	0556	1517	0426	0114	0000	0003	0157	0000	0000	4014	0057
033510	1411	1511	2423	0000	0000	2001	2323	2717	2200	0000	4014	0000	0000	0001	2425	2022	1706	1114	0500	0001
033514	0112	0000	0000	4015	0020	4044	0000	0000	0000	2202	2306	2300	0000	0000	0000	0000	0000	0000	4015	0065
033520	0535	3333	0320	0000	0030	0000	0000	0000	4016	0004	0401	2401	0405	0600	0000	0000	0000	0000	4016	0054
033524	0401	2401	1501	2600	0000	0000	0000	0000	4017	0042	0402	0617	2215	0000	0000	0000	0000	0000	4020	0002
033530	1324	2304	1520	0000	0000	0067	0000	0000	4020	0146	1411	0224	0123	1300	0000	0137	0000	0000	4021	0062
033534	2022	0523	1115	0000	0000	0500	0000	0000	4022	0004	2422	0116	0530	0000	0000	0000	0000	0000	4023	0006
033540	2422	0116	0530	3400	0000	0000	0000	0000	4023	0143	2422	0116	0530	3500	0000	0000	0000	0000	4024	0033
033544	2422	0116	2311	1500	0001	0253	0000	0000	4024	0042	4004	0000	0000	0000	0132	0204	1514	1102	0001	0000
033550	0000	0000	0000	0000	3675	2422	0116	1411	0201	0000	0000	0000	0000	2000	4016	2324	1115	2514	0100	0000
033554	0160	0000	0000	4025	0056	2405	1405	3000	0000	0000	0000	0000	0000	4026	0063	2405	1405	3034	0000	0000
033560	0000	0000	0000	4026	0124	2405	1405	3035	0000	0000	0000	0000	0000	4027	0030	0214	0116	1300	0000	0001

27-21

27-22

MTR dedicated PP

PP 00

00XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	7145	7413	----	----	7777	----	----	0006
X1	1003	7113	----	----	7776	1405	----	----
X2	0002	6500	----	----	----	----	----	----
X3	----	0001	----	----	1600	0274	----	----
X4	----	----	----	0350	----	5771	----	----
X5	----	----	6600	----	0031	6055	6231	----
X6	0006	----	6700	----	0024	0632	6311	----
X7	0001	----	6440	0001	0030	----	0031	----

01XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	4401	2300	3020	0516	1605	6532	1003
X1	----	3601	6746	1006	3020	6010	3020	1601
X2	----	2300	0572	0703	1005	3010	1007	6010
X3	----	6650	1400	0100	2200	1202	0604	3011
X4	2000	572	3401	0230	4000	1102	3021	1007
X5	6532	1400	3025	3023	0511	1012	5401	0633
X6	3401	4401	3101	2200	3024	3321	6572	1501
X7	1500	3631	6020	4000	1003	5401	3024	3402

02XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1402	0567	3021	3601	5001	4177	1400	0600
X1	3502	3021	1071	1120	6572	5400	5400	1601
X2	5002	1277	0406	0403	1016	4223	1417	5400
X3	0276	3403	3403	0100	0773	1405	2300	1753
X4	0424	5002	5002	0125	3601	6010	0500	0100
X5	3323	0277	0277	1420	2300	3012	5400	0315
X6	2200	5403	5403	3401	1100	3210	1747	0402
X7	3777	6612	6612	3701	5400	2300	2300	1740

03XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0403	0411	6060	6060	6060	6060	6060	6060
X1	----	3010	3060	3060	3060	3060	3060	3060
X2	0404	1504	0403	0403	0403	0403	0403	0403
X3	----	----	0100	0100	0100	0100	0100	0100
X4	0406	----	1521	1554	1554	1554	1554	1554
X5	----	2000	2000	0310	2000	2000	0310	0310
X6	0410	6211	6221	6231	6241	6251	6261	6271
X7	----	3466	3465	3466	3466	3466	3466	3466

04XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	6060	6060	6060	6060	6060	6060	6060	6060
X1	3060	3060	3060	3060	3060	3060	3060	3060
X2	0403	0403	0403	0403	0403	0403	0403	0403
X3	0100	0100	0100	0100	0100	0100	0100	0100
X4	1554	1554	1554	1554	1554	1554	1554	1554
X5	2000	2000	2000	2000	2000	0310	2000	2000
X6	6301	6311	6321	6331	6341	6351	6361	6371
X7	3466	3466	3466	3466	3466	3466	3466	3466

05XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	6060	6060	6060	6060	6060	3437	5256	2377
X1	3060	3060	3060	3060	3060	2000	0001	7777
X2	0403	0403	0403	0403	0403	0100	0603	3154
X3	0100	0100	0100	0100	0100	6370	2101	5456
X4	1554	1554	1554	1554	1554	6376	----	0001
X5	0310	0310	2000	2000	3037	0200	5256	5256
X6	6401	6411	6421	6431	0407	5225	0002	0003
X7	3466	3466	3466	3466	1400	3054	0725	3474

06XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	3047	5456	0704	1750	0010	1000	3255	3600
X1	3457	0003	2001	----	0200	5400	0603	3002
X2	4056	0200	0626	1147	1024	0100	2101	2177
X3	5400	5225	3456	5760	5760	5245	----	6027
X4	0606	1404	0100	0010	0010	1400	2177	0672
X5	0200	3556	0315	5400	5200	3400	6027	3001
X6	1041	2177	0740	1041	1335	7014	0763	3202
X7	3074	7125	5610	5760	5000	3401	3402	3455

07XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	3000	0106	1413	0405	0607	0653	1424	3013
X1	3554	6250	0200	2001	3651	1401	6010	5400
X2	1063	0100	5343	0013	1063	0200	3011	0634
X3	0404	0652	2001	0200	3550	5343	5400	3014
X4	3553	6010	5060	5343	2000	0304	0644	5400
X5	1063	3011	6010	0100	0106	0001	3012	0640
X6	3552	3112	3011	0652	6250	0200	5400	1440
X7	2000	0404	3112	0100	0200	5343	0650	6010

10XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	5600	1400	0100	3030	0100	0471	1460	3007
X1	1022	5400	0737	2200	0607	3374	6010	3210
X2	3214	1022	----	2000	0200	1622	3012	0707
X3	0605	1454	0100	0471	1451	6010	3374	3074
X4	3012	0200	0607	1446	0473	3014	0455	1071
X5	0411	6251	0200	0200	3030	3405	3012	3422
X6	3251	0200	1451	6251	2200	3010	1622	1412
X7	0607	1416	0473	0363	4000	3407	6010	0200

11XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	5343	3012	3422	----	2200	0200	1237	0355
X1	0100	1622	1412	----	7377	1451	0510	3030
X2	1040	6010	0200	----	2300	0473	3012	2200
X3	1461	3007	5343	----	0400	3074	0406	7000
X4	6010	3210	0100	----	3414	1622	1202	0451
X5	3012	0707	1040	0200	0100	6010	0560	3074
X6	3374	3074	0304	0306	0100	3030	0200	1621
X7	0455	1071	0321	3014	0607	3407	6152	6003

12XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1627	3030	0306	3011	3074	1303	3213	3512
X1	6020	1011	1401	2177	1623	1063	0744	3074
X2	3020	0707	3411	7677	6010	3401	1014	1623
X3	0425	1401	1440	0772	3012	3022	3124	6210
X4	3006	3412	0200	3011	1006	3201	3214	3006
X5	1277	3074	6251	2177	0760	0750	0740	1277
X6	1103	1622	0100	0017	1006	1014	2000	1705
X7	0512	6210	1146	0666	5400	3123	4000	0625

13XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1605	6020	6220	3422	0473	3030	3111	3010
X1	1014	3011	1454	1402	3031	1011	0544	0444
X2	2303	3221	0200	0200	0520	0610	3030	3011
X3	3001	7611	6251	5343	3074	3033	1240	1337
X4	0200	3011	0100	0100	1622	1006	0551	3411
X5	6362	3421	1146	0607	6010	1601	3074	3074
X6	3074	3074	3074	0205	3012	6010	1625	1070
X7	1622	1622	1071	1451	0566	3010	6010	3511

97404700A

97404700A

DSD dedicated PP

PP 01

00XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0003	0001	0001	5534	5542	----	----	0001
X1	1500	----	7773	4157	3650	----	----	0100
X2	7301	----	----	3434	3344	0311	----	1000
X3	----	2617	----	5735	5033	1120	----	0003
X4	7776	0011	----	3557	4457	1326	4770	----
X5	5230	7145	----	5605	0030	6700	----	6210
X6	0032	6000	0024	0733	0026	7700	----	6211
X7	----	7263	0002	----	5400	6440	----	6212

01XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	7575	1314	0773	0317	1671	6200	----	0403
X1	5700	7356	7233	2516	5700	0157	0201	1671
X2	----	7157	4156	2456	4136	----	6200	5700
X3	5111	----	7157	7157	6241	0212	0403	4136
X4	7301	----	----	----	0125	6250	1071	6216
X5	0304	5132	----	0202	2417	0214	5700	0516
X6	0607	7303	5153	6235	5700	1124	4136	7700
X7	1012	0406	0103	0103	0144	3257	6206	0131

02XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	6200	0603	5700	1101	1601	5700	1706	0201
X1	0522	1671	4136	1671	1603	4136	0671	6270
X2	2214	5700	6224	5700	0557	6271	5700	2125
X3	1707	4136	1104	4136	----	1701	0201	0525
X4	5671	6224	1405	6247	0144	1671	6303	0556
X5	----	0616	5700	1501	6200	5700	1716	7400
X6	0202	0371	0201	1116	1503	4136	7157	0210
X7	6237	5671	6200	2405	1071	6260	----	6200

03XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	2324	5700	5413	1417	5700	6200	3057	0610
X1	0520	4144	2411	0313	4000	3057	7201	7356
X2	5700	6231	1505	5700	5403	7201	3270	7700
X3	4000	2331	5777	----	3057	3270	5174	0212
X4	5365	2307	----	5401	7201	5671	----	6200
X5	2324	1757	4217	2516	3270	5700	0202	7303
X6	0520	----	6200	2324	5700	0202	6267	0473
X7	5671	----	2516	0520	0202	6200	7301	7700

04XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0132	6200	6200	0411	5700	5355	7217	4000
X1	6200	7320	7157	2357	----	7157	4777	5266
X2	7311	2223	7201	----	5326	7205	----	7156
X3	1314	2425	0377	0140	7157	1677	0130	5671
X4	1517	2627	----	6402	2324	----	6200	5700
X5	7377	7377	0127	7157	1720	0127	7156	4000
X6	----	----	6200	0422	5700	6200	7157	5266
X7	0133	0134	7157	1720	----	7157	----	7156

05XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	7233	4673	5671	3756	0474	----	0200	2220
X1	3756	7157	5700	7157	----	6510	1272	3446
X2	7157	----	4135	----	4135	0554	0200	1007
X3	----	4135	6200	4135	6232	7510	1407	3447
X4	4000	6200	7173	6200	4444	0200	7710	2000
X5	5266	7173	5645	7173	5700	2254	7000	6453
X6	7173	5645	4673	5645	----	0200	7410	3415
X7	5645	4673	7233	4673	5257	1221	5[illegible]26	0200

06XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	6462	2220	7154	5614	0651	0100	3235	1431
X1	7510	3446	7510	2300	0200	0641	2200	6240
X2	0200	1007	3055	2001	1127	2000	7777	1427
X3	1470	3447	1601	5400	0200	0106	2177	6250
X4	7710	2000	6010	4035	1117	6010	6027	1426
X5	7100	7145	3014	0200	3636	3014	0613	6370
X6	7410	3415	1201	3470	0100	2101	1430	6120
X7	5027	0200	5400	0200	0551	----	6230	0350

07XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	2377	1745	0745	1745	3454	1746	1130	2000
X1	7777	0754	1404	0766	3653	0705	0403	3357
X2	3114	2000	3554	1466	2000	2000	0100	3431
X3	3435	3357	2000	3532	3333	3357	1025	5600
X4	3654	3434	5733	1071	3432	3431	1450	6124
X5	3071	3633	3433	1741	3071	3630	3553	3071
X6	3534	1277	3632	0761	3531	3053	1502	3544
X7	1071	1741	1277	1400	1071	1277	3530	1071

10XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1745	6124	6123	3652	1277	0403	3071	6123
X1	0720	1071	3053	2000	1745	0100	3541	1071
X2	2000	1745	1071	3457	0703	0666	2000	1745
X3	3357	2000	2300	3444	1466	3071	3334	0721
X4	3444	3333	0037	2000	3542	3552	5400	2077
X5	3643	5400	0524	5033	3042	2000	6124	6577
X6	1466	6124	3071	3443	2300	3334	1475	5500
X7	5500	5600	3453	3642	3436	3442	5500	6123

11XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	5600	0704	1402	2000	2000	0555	0200	1403
X1	6122	2000	3401	6610	0002	3045	0651	0200
X2	2000	5533	1444	6010	0563	3446	2001	5654
X3	3350	3445	6101	3010	7710	1410	----	3012
X4	3441	0100	6474	0504	7020	3411	1701	0463
X5	3640	[illegible]666	0370	5400	7410	1404	0576	5000
X6	1277	[illegible]100	0100	1141	7010	0200	1410	1131
X7	1745	[illegible]645	0643	0366	7510	5654	3411	6010

12XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	3010	7510	0100	3402	50[illegible]1	3702	3603	7770
X1	1237	5400	0560	1402	2220	0657	1102	0100
X2	5400	3626	1400	7310	04[illegible]6	7510	0403	0562
X3	1141	1424	3403	1267	3245	5000	0100	7710
X4	7710	5400	7710	5003	06[illegible]4	1225	1224	7001
X5	7020	3635	7001	0026	3145	3371	0100	2000
X6	7410	0100	7410	3401	0200	5400	1220	7020
X7	7010	1126	1402	7210	3363	1225	6260	3417

13XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1403	[illegible]711	1347	3501	3063	3116	3501	1404
X1	3402	7710	7410	5001	0421	7210	0364	3601
X2	2000	7000	1402	4765	3036	5001	3061	4001
X3	1004	1404	7310	1006	1214	4765	0412	7210
X4	5400	3517	0[illegible]16	5101	1075	1006	1701	0574
X5	1347	1502	1501	4766	3401	7210	3401	3702
X6	3060	3502	3401	7210	3001	0404	1402	0703
X7	1741	5700	1402	0567	1004	1404	7310	0100

27-24

ITS pool PP

PP 02

00XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0003	----	----	0020	0001	3424	----	0001
X1	1500	1405	0061	1405	0017	2327	3761	0100
X2	7302	----	2401	----	----	----	----	1000
X3	----	0274	0041	5771	----	----	----	0003
X4	2033	5771	0003	0016	----	0002	----	5600
X5	----	1073	1000	0020	0001	2775	----	6220
X6	4202	3001	----	0020	----	0003	----	6221
X7	0107	2772	5261	0020	0030	----	----	6222

01XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1477	3474	0135	1014	0326	0200	1464	0533
X1	1701	3051	0200	3102	1063	0335	6170	5400
X2	0576	1377	0424	6114	2300	0552	7774	0275
X3	3075	1006	0115	1073	2300	2002	5400	5000
X4	6050	3350	0005	0100	5400	1401	0272	0135
X5	3051	0462	5400	0121	0325	0200	1702	5400
X6	1237	1006	0133	1014	2001	0335	3417	0301
X7	1007	0200	3001	5400	3014	0504	5000	5000

02XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0325	0200	0313	3406	0001	3407	4417	0446
X1	5400	0364	2000	1400	3403	3016	0605	2000
X2	0305	3076	0477	3407	3017	4417	3001	1073
X3	5000	6003	3517	0200	0200	3617	0503	3415
X4	0326	0200	3007	0606	0616	4017	0100	2000
X5	5400	0547	1006	4017	1057	1071	0221	1104
X6	0306	5600	0607	3416	3401	1021	3004	5400
X7	1422	0576	1071	5017	4017	3303	0200	0533

03XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	2000	0134	3011	3012	1063	0001	6010	3010
X1	0121	1401	1377	0502	6010	6057	3010	1736
X2	5400	3502	1006	3015	3011	----	0571	0761
X3	0135	1063	3310	3415	3402	----	0100	2001
X4	1400	3501	0410	0100	3010	1477	0474	7171
X5	2323	1014	2323	0157	3401	1701	3410	6010
X6	3424	3102	3424	5400	0346	0576	3076	3010
X7	0101	6010	0561	0311	----	3076	6210	3111

04XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0572	6010	3456	3210	3411	1404	6312	3411
X1	2001	3014	3013	0562	1412	0200	0573	1406
X2	7163	0441	3455	1411	0200	0364	2034	0200
X3	6370	0371	0100	0200	0364	0371	0404	0364
X4	0350	3074	3000	0364	0371	2001	0200	3011
X5	2001	1620	1457	0356	0100	4620	0533	0556
X6	7163	6010	6010	0100	0271	6373	2000	4001
X7	2610	3014	3074	1725	3411	0001	----	0507

05XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0100	1410	0571	0557	5415	2000	0600	0200
X1	2263	3400	3602	0100	0006	6600	5400	0135
X2	3401	4001	6210	0466	0115	3105	0576	0355
X3	1063	4400	3277	1104	0007	6010	0420	----
X4	5400	0402	3412	0200	3011	3013	5300	1100
X5	0467	3601	1703	0135	3404	2200	0600	----
X6	3076	3600	0403	5000	0100	3777	2341	0001
X7	3402	1115	3014	0533	0216	5300	----	----

06XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0411	1011	0633	0502	0653	0502	3402	2200
X1	0153	0100	0200	7420	0200	7420	7720	2012
X2	0153	1460	0702	7120	0702	7320	0012	0527
X3	0100	0324	5600	2771	5600	1463	7420	2042
X4	1017	2400	0716	0520	0716	0517	7020	0411
X5	0100	0100	7720	1404	7720	6620	5400	0200
X6	0235	0245	0004	0322	0005	0655	0573	0533
X7	0100	5400	2000	5400	2000	7520	0410	0102

07XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0611	2000	0460	3610	0405	5400	3513	3413
X1	0100	0106	1500	3013	7720	0736	3006	3612
X2	0623	2300	3412	2177	0010	5100	2200	3013
X3	5600	0153	3410	7624	6420	0577	3776	1730
X4	0711	0405	3007	0672	0743	3410	1020	0673
X5	1400	2000	5400	2000	7720	3013	3411	3006
X6	5400	107	0716	----	0002	5400	1056	1021
X7	0575	3307	3413	3310	3010	0711	3113	0603

10XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1411	0655	0200	3401	0571	0652	0745	0030
X1	3512	1502	0364	0303	0100	0655	1002	0031
X2	7420	5400	3077	4502	0544	0657	1004	0032
X3	1404	0716	6170	3601	0625	0661	1037	3424 IT
X4	7320	1400	0576	4001	0631	0663	0024	2300 S
X5	0010	0100	6010	3402	0632	0664	0025	1073
X6	1470	0605	2000	3011	0645	0741	0026	----
X7	0100	1420	1043	1720	0651	0743	0027	0311

11XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0200	0603	0494	----	6611	7777	0406	2000
X1	2174	2101	3011	3074	1146	0512	0100	1220
X2	7411	----	3431	1620	1477	0200	1103	0200
X3	2000	3434	1401	6010	1701	1375	2000	0501
X4	0106	3014	3432	3011	0576	2000	1242	1400
X5	6010	3433	3035	0531	0363	----	0305	6010
X6	3014	3011	3436	6511	7011	6010	0200	3614
X7	3233	3331	3077	1215	2300	3030	1253	3053

12XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	1377	1444	2324	1405	2400	5700	0200	3001
X1	1006	0200	1115	2405	----	----	0424	2127
X2	3352	0364	2514	----	1725	0100	6711	7503
X3	0405	0100	0124	2405	2455	1252	1256	6020
X4	3155	0103	1117	1405	1706	1400	7011	3020
X5	1006	2000	1655	3055	5523	3403	1400	1006
X6	1711	1233	0317	0102	3116	6511	3401	0721
X7	6210	0352	1520	1722	0310	1215	3402	3002

13XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0520	3001	7211	0403	1400	3072	3040	7011
X1	3603	2127	6611	0100	3425	3425	2127	3426
X2	1720	7503	1321	1256	0316	0306	7503	3736
X3	0511	6220	7011	7511	1400	3072	6220	0503
X4	3403	3072	3631	0100	3425	3425	3640	0100
X5	2000	3402	3335	1252	0317	0307	3335	1154
X6	4000	0302	0541	5440	5440	5440	0502	3025
X7	3420	1400	3002	2176	2176	2176	3440	7211

97404700A

97404700A

ITD pool PP notice that PP res has been destroyed,

PP 05

00XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0003	----	0035	0166	----	2005	----	0001
X1	1500	----	0035	2711	----	----	5770	0100
X2	7305	----	----	1100	----	----	----	1000
X3	----	----	----	----	----	0043	0003	0003
X4	5555	5730	2223	----	----	0023	0002	0206
X5	----	4700	4471	0215	----	1441	----	6250
X6	----	1073	----	----	5730	1406	----	6251
X7	4251	7277	----	0001	----	6713	----	6252

01XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

02XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

03XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

04XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

05XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

06XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

07XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

10XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	----	----	----	----	----	----	----	----
X1	----	----	----	----	----	----	----	----
X2	----	----	----	----	----	----	----	----
X3	----	----	----	----	----	----	----	----
X4	----	----	----	----	----	----	----	----
X5	----	----	----	----	----	----	----	----
X6	----	----	----	----	----	----	----	----
X7	----	----	----	----	----	----	----	----

11XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0100	6010	0702	5014	3003	0603	0576	1066
X1	1100	3014	3563	6000	3404	2101	2000	1017
X2	0375	3261	3074	3004	1414	----	0106	3313
X3	2745	0603	1620	0404	2103	3201	6010	1003
X4	0100	2101	6000	1701	5000	0611	0361	2200
X5	6563	----	3001	0576	6200	2000	3014	7777
X6	2000	3464	0555	0363	3014	0175	5400	5200
X7	0106	3263	2103	3437	3261	1701	0061	1103

12XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	0404	3077	2000	7716	4502	1237	1305	7611
X1	5500	6260	7461	2100	3601	0403	1701	1477
X2	1103	2000	3455	6600	4001	0100	3456	6511
X3	1401	0100	5137	6010	3402	1357	3014	1301
X4	3462	5400	7730	2000	3011	3447	2200	1701
X5	1402	1305	5400	1442	1711	5037	7000	0574
X6	5400	1403	1524	3401	0571	7727	3403	7511
X7	2041	3454	5037	0303	3014	5500	1101	0100

13XX	0X	1X	2X	3X	4X	5X	6X	7X
X0	6551	1307	7111	6010	0100	5037	1406	0100
X1	5037	7511	7461	3045	1072	7716	3456	5400
X2	7730	3003	0554	3213	0405	0403	2000	1522
X3	7411	1103	7511	1014	1446	0100	1441	2000
X4	7311	7611	0200	3146	0200	1220	3455	1403
X5	0100	5037	1527	3214	7566	----	1500	5400
X6	0570	7730	2004	0603	3660	1106	4456	1523
X7	6611	7411	2647	2100	3637	2000	2000	5037

QUESTIONS/ANSWERS SETS 28

28.0 INTRODUCTION

This section consists of 15 question/answer sets. Each question set is identified by number together with the appropriate section in this manual to which it relates.

The answer sets follow the last question set.

28.1 QUESTION SETS

Obtain a current listing of the following areas which will be used for some of the Question Sets.

1. Catalog of the system
2. Dump of CMR. (Use program listed in Section 23.5 or that shown in Section 27.)
3. SYSTEXT (PPCOM, CPCOM)

QUESTION SET NO. 1 (CMR Section 2)

Obtain a dump of a current system CMR (Use the program shown in Section 23 to obtain one.) Answers are based on the dump in Section 27.

1 For this system -

   a. How many PP's?

   b. How many control points?

   c. How much central memory?

   d. Does the CEJ/MEJ option exist?

   e. How long is CMR?

2 Is there a PP program running in PP3?, PP9? If so, which program?

3. Are any of the PP's making MONITOR requests?

4. Describe the peripheral equipment configuration for this system.

5. Which device holds the system files? Permanent files?

6. Locate the file SYSTEM, i.e., which equipment and which track does the file reside on? Where is the file positioned?

7. Identify File Name, File Type, Control Point assignment, equipment, and file positions at the following FNT locations;

   6704, 6714, 6726, 6734

8. How much CM is available?

9. What is the job switch (CPU SLOT TIME) delay?

10. What is the time of day?

11. Which control point has access to the CPU?

12. What is the original input queue priority for a BATCH job?

13. What CPU priority is assigned to an EI200 job?

14. What is the ROLLIN queue priorities for TELEX origin jobs?

15. CIO is a PP program residing in the RPL. Find its RPL entry.

16. What is the name of the program following CIO in the RPL.

17. What are the base addresses of the PLD, RCL, and CLD? What are the names of the 1st entry in each library or directory?

18. Where does the system dayfile begin? The dayfile dump buffer?

19. Which PP has access to channel 0? Channel 3? Channel 10B? Channel 20B?

20. What is the first unavailable channel?

21. The following questions refer to control point 5: (or any active control point)

    a. What is Jobname and Origin?
    b. What is the control point status?
    c. How many PP's assigned? Which ones are they?
    d. What is its RA and FL?
    e. What are CPU and QUEUE priorities?
    f How much CPU time has been accumulated?
    g. What is the user number and user index for this job?
    h. How many sectors of mass storage have been transferred?
    i. What is the first control card in the control card buffer?
    j. What is the next control card to be executed?

22. Which PPs have been locked out (turned off) by the system and how did you arrive at the answer.

23. Why can't the FNT table start or end beyond location 4096D?

QUESTION SET NO. 2 (MONITOR Section 3)

1. Explain system interaction, i.e. the means of communication between POOL PROCESSORS, MONITOR, and a CONTROL POINT.

2. What XCHG packages exist with

   a. CEJ/MEJ option

   b No CEJ/MEJ option

3. What does a pool processor do to make a CPUMTR request

   a With CEJ/MEJ option

   b. Without CEJ/MEJ option

4. How does the PPMTR make a CPUMTR request

   a. With CEJ/MEJ option

   b. Without CEJ/MEJ option

5. How does a CONTROL POINT make a system request

   a. With CEJ/MEJ option

   b. Without CEJ/MEJ option

6. How does the CPUMTR decide which control point to activate? If a new control point is activated, how does CPUMTR manage the exchange packages?

7. What is the difference between CPUMTR program mode and monitor mode?

8. When in monitor mode, what does the "XJ" do? In program mode?

9. What section of the CPUMTR will be activated for a pool processor request? A PPMTR request? How is this accomplished?

10. Explain what each of the PPMTR timed scan processors accomplish?

11. What does the PPMTR do when it detects that a control point has exceeded its CPU TIME SLICE?

12. How does PPMTR take a control point out of periodic recall?

13. Who processes all RA+1 requests?

14. What criteria does the PPMTR use when switching control points?

15. Basically, how does the CPUMTR determine which control point should get access to the CPU?

16. Devise a scheme for using two sub-control points. Tell how you would implement it.

QUESTION SET NO. 3 (PP RESIDENT Section 4)

1. Where must a PP program be coded to run, if it is to interface with PPRES?

2. Why is it desirable to have a pool processor pause for storage relocation?

3. What is a pool processor doing when it is waiting for MONITOR to assign it to a job?

4. How does a PP program make requests to MONITOR? How does it know if the request has been honored? How does PP know when request is completed?

5. How do two PP's keep from getting the same channel?

6. How does a pool processor program decide which control point it is attached to?

7. How could a PP program write a message to the control point dayfile?

8. After pausing for storage relocation, where will a PP program find the updated RA and FL for the control point it is attached to?

9. Since some PP programs will not fit in a PP, individual functions may have to be written as overlays How can a program get one of these overlays loaded?

10. How is a mass storage driver loaded?

11. What 3 entry points exist in all mass storage drivers?

12. Why should a programmer be very careful about using location 7000 - 7777 when dealing with mass storage I/O?

13. Design and flowchart a PP program to list in the control point dayfile all common file names as found in the FNT.

QUESTION SET NO. 4 (JOB PROCESSING Section 6)

1SJ - Job Scheduler

1. When is 1SJ called?

2. How is 1SJ requested?

3. How do the queue priorities get aged? What routine actually does the aging?

4. How is the CM TIME SLICE checked? What happens if a control point has exceeded its CM TIME SLICE?

5. Can you disable priority evaluation? Job scheduling? Auto rollout?

6. Under what conditions might the job scheduler request that a job be rolled out?

7. What criteria does 1SJ use to determine the "BEST" job for scheduling?

8. When does 1SJ call 1AJ? 1RI?

9. What does 1SJ do if MONITOR indicates that a PP doesn't exist for 1AJ or 1RI?

1AJ - Advance Job

10 Why does PPMTR call 1AJ?

11. What are the 3 main overlays that 1AJ calls?

12. Explain why the overlay 3AA, BEGIN JOB, is called and what it accomplishes.

13. Does the BEGIN JOB overlay ever get called for a TELEX origin job?

14. Why would the overlay 3AB, PROCESS ERROR FLAG be called? What processing does it do?

15. Which 1AJ overlay is called to process the next control card?

16. Which control cards will be processed entirely within the 1AJ PP?

17. Where are the majority of most of the control cards processed? How does control get to these routines?

18. What does 1AJ do to process a program call card which refers to a user's file referenced in the FNT?

19. How does 1AJ load an absolute CP program that is referenced in the CLD?

20. How does 1AJ process the KRONOS control language statements?

21. Is it legal to call a PP program from a control card? If so, how does 1AJ process it?

22. Write a CP job that can read absolute memory. Include all control cards and use a DMP- special entry point.

23. What will the dump contain if you include a DMP (field length) card immediately after executing the job written for 22.

24. Explain the sequence of events when the call card for the job in 22 is encountered. Continue the explanation when the DMP card is encountered.

25. What words in CMR are used to process sub system initiation?

26. What does 1DS do when a sub system is initiated?

27. What is the OUTPUT queue? What is the ROLLOUT queue?

28. How does a rollout file differ from a DM* file?

29. Why isn't the ACCOUNT card aborted by the system?

QUESTION SET NO. 5 (SYSTEM I/O Section 7)

1. Explain the purpose of the PP Resident routine SET MASS STORAGE (SMS).

2. Explain the system table linkage in accomplishing I/O from a central processor program.

3. Why do you specify a sector count when requesting mass storage space?

4. What subroutine within a mass storage driver would you use to read a sector? Write a sector? Position a mass storage device?

5. Why is the last operation when writing a mass storage, a DROP TRACK monitor function?

6. Go through the system table linkage for any file you wish using the CMR dump you generated. What is this files length in sectors? (For the CMR dump included use file STIMFL. Where is the FNT located? Describe it, and trace table linkage.)

QUESTION SET NO. 6 (CIO Section 8)

1. If CIO gets a request to process a non-existent file, what happens?

2. Does CIO need help from any other PP to do mass storage I/O? Explain what happens during a mass storage read/write operation.

3. How does CIO handle random I/O? Explain what happens during a random READ operation.

4. Explain the term re-write in place.

5. Under what conditions will CIO drop out, while accomplishing a mass storage read operation?

6. Explain what happens when CIO gets a read request for a magnetic tape

7. What happens when CIO gets a READ/WRITE request directed to/from a TTY type equipment?

8. If you were to add another I/O driver to the system, what changes must you make to CIO?

QUESTION SET NO. 7 (MAGNET & RESEX Section 9)

1. Name some of the control cards processed by RESEX?

2. Since RESEX is a CP routine that runs at the user's control point, how does the system avoid destroying the user's job when loading RESEX? (Briefly)

3. What is the purpose of the PREVIEW display?

4. Only one routine updates information for the PREVIEW display Which routine?

5. The FST entry for a magnetic tape file contains a UDT address. Where are UDT entries stored?

6. Briefly, what is the purpose of the RESOURC control card?

QUESTION SET NO. 8 (FILE MANAGERS Section 10)

PERMANENT FILE QUESTIONS

Refer to the dump created in Question Set 1 where necessary.

1. What is the first track of the INDIRECT file chain? The PERMIT buffers?

2. How many tracks are allocated to INDIRECT permanent files: Where is this information kept for each user?

3. How many tracks are allocated for PERMIT buffers and catalog tracks?

4. Do any DIRECT access permanent files exist?

5 What happens when an INDIRECT access permanent file is purged?

6. How can PFM keep track of "holes" in the INDIRECT file chain?

7 Assume user number USER001 has permanent file XYZ. What implied permission will user number USER*** be granted?

8. What is the difference between a SEMI-PRIVATE and LIBRARY file?

9. Must INDIRECT access permanent files reside on a user's MASTER DEVICE? DIRECT ACCESS?

10 Is multi-read access possible with DIRECT access permanent files? If so, how is it implemented?

11. Why is multi-read access no problem when dealing with INDIRECT files?

12 What happens when you ATTACH a DIRECT access file and have WRITE permission? If one user attaches a file with WRITE permission, can another user attach this file? Why?

13. What happens when you PURGE a direct access permanent file?

14. What happens when you DEFINE an existing file that resides on a device not configured to hold permanent files? Could the above situation ever occur?

QUESTION SET NO. 9 (LOADER and Binary Deck Formats Section 12)

1 What difference exists between OVL and ABS binary deck formats?

2. Why is ABS format desirable for utility program routines?

3. What is the 1st table in any binary deck?

4. Which KRONOS program processes relocatable binary decks? How is this program loaded?

5. What does the KRONOS relocatable loader do when it detects:

   a. An OVL or ABS binary deck table.

   b. A loader control card.

6. Following the label table, what must be the 1st table of a relocatable binary deck? What is this table used for?

7 What control cards does the loader read and process?

8. How are USER LIBRARIES utilized?

9. Can a user specify his own user libraries? If so, how?

10. What PP program is called as a result of an OVERLAY system macro? How does it accomplish its task?

11. What usefullness is ASR? How would you put a PP program on an ASR? How can ASR be easily eliminated?

12. Why is it important to ensure that all the SCOPE 3.4 libraries are at the same PSR level?

QUESTION SET NO. 10 (TELEX Section 13)

1. Which CPU and PPU programs make up the TELEX INTERACTIVE subsystem?

2. Describe in general the TELEX control point.

3. How is the dynamic memory associated with TELEX, i.e., POT3 managed?

4. Explain in general the TELEX origin job flow for each of the following:

   a. Job initiation
   b. Terminal input
   c. Terminal output

5. What is the interface between TELEX and 1TD - the 6676 driver?

6. How do PP programs 1TA and 1TO interface with TELEX?

7. What is the TELEX re-entry table used for?

8. Name and explain each of the TELEX internal queues.

9. Where is all of the current information about any given terminal kept?

10. What is a multi-terminal job?

11. How do the terminals get processed as TELEX progresses around its main loop?

12. What happens when TELEX requests a PP thru the CPUMTR and no PP's are available? Is this handled as a special case for TELEX? Why?

13. How does TELEX get activated?

14. Why does TELEX queue up a group of requests for 1TA?

15. Why are all initial TELEX jobs scheduled to the rollin queue instead of the input queue?

16. Why does 1TA create a dummy rollin file for each terminal at log-in time?

17. What happens when a TELEX origin job terminates?

18. What function does 1TO accomplish for TELEX?

19. Explain why an output is followed by an input in 1TD.

20 Explain the re-entrant concept in 1TD.

QUESTION SET NO. 11 (EXPORT/IMPORT Section 16)

1. Why does EI200 utilize a control point? What prevents this control point from being rolled out? Why does E200CP get the CPU "whenever" it wants?

2. What tables are associated with each terminal?

3. How does 1ED, the 6671 driver, communicate with 1LS, the EI200 executive?

4. Who initiates EI200CP? Why? How?

5 What characteristic of the 6671 driver, 1ED, enables the 200 U.T. user the ability to SUSPEND a file which is currently being printed?

6. Which function does the EI200 SERVICE PROCESSOR-XPS accomplish? Who calls XPS?

7. How does EI200 accomplish mass storage I/O?

8. Trace the flow of data from a 200 U.T. card reader to the INPUT QUEUE.

9. Trace the flow of data from the OUTPUT QUEUE to a 200 U.T. line printer.

10 How does EI200 determine which terminal a file in the OUTPUT QUEUE should be routed to?

QUESTION SET NO 12 (BATCHIO Section 17)

1. What 3 PP programs are involved in the BATCHIO subsystem?

2. How is the BATCHIO control point utilized?

3. Why can't the BATCHIO control point ever be rolled out?

4. What is happening when BATCHIO is in its "IDLE" state?

5. How does 1CD, the BATCHIO UNIT RECORD EQ. DRIVER, do I/O to/from mass storage?

6. What does BATCHIO do to submit a job to KRONOS?

7. How does BATCHIO utilize the ID field in an FST entry?

8. What function do the DCW words serve?

9. How is the DRQR one word request stack utilized? How does 1CD know the request is for him?

10. How does 1IO know when to call its preset routine?

QUESTION SET NO. 13 (DEADSTART Section 24)

1. Why are all the PPs A register set to 10000B?

2 What are each of the PPs doing at deadstart time?

3. Why are the position of the first 10 PP routines on the deadstart tape important?

4. What happens to the information represented on the deadstart panel at deadstart time?

5. What does SYSEDIT do at deadstart time?

6. Explain the importance of PPULIB?

7. What does STL do before dropping out? Why is this last step necessary?

8. What is the philosophy of recovery in KRONOS V2 1?

QUESTION SET NO. 14 (DSD and DIS Section 25)

1. When does DSD take on the attribute of being attached to a Control Point?

2. How does DSD use 1DS?

3. Do the overlays actually display the buffer? If not, what does?

4. Explain the sequence of events when the console operator types X.DIS (CR) (carriage return)?

5. How do DSD and DIS both use the console display device?

QUESTION SET NO. 15 (INSTALL Installation Handbook)

1. Given the common file MODS (see following example) modify PFDUMP, 1RO, and 1TA and:

   a. Modify the running system

   b. Build a new deadstart tape

```
*IDENT PFDUM7
*DECK PFDUMP
*I,463
          NZ       X1,GRL2.2       IF NOT MASTER DEVICE
          SA1      ORDNO
GRL2.2    BSS      0
*/        ****     TO HALT CANCEROUS FNT GROWTH
*DECK 1RO
*D,KRON14.1,KRON14.2
*I,1098
          LDN      0
          STD      CM+1
*/        ****     PRIMARY FILE FNT MASHED ON RECOVERY.
*/        ****     ALSO NULL FILES NOT VALIDATED ON RECOVERY.
*DECK 1TA
*D,1645,1651
*I,3526
          LDD      BA
          LMC      BFMS+2+TTSS+VFST*5
          ZJN      VFLO            IF PRIMARY FILE
          LDM      2,BA
          ZJN      VFLX            IF FILE NOT WRITTEN ON
VFLO      BSS
0000000000000000000000
```

2 Write a procedure file to modify and to create a new user library

3. Write a job to add the above procedure file to a new deadstart tape.

QUESTION SET No. 15 (continued)

Write out a CMRDECK for the following system configuration:

Mass Storage Equipment

1. 6638 with Unit 0 on Channel 0 and Unit 1 on Channel 1. Both units are to be configured to hold "Temporary Files". Unit 0 is to be the "System" device when no "SYSTEM" entry is typed during Deadstart.
2. Four 844's configured as two dual-access devices with Units 0 and 1 as one device and Units 2 and 3 as the other. They will all be accessed via Channels 2 and 3.

   These two devices make up the default Permanent File Family named "FAMX" Assign Family Device Numbers 40 and 41. Both are to hold Direct Access Permanent Files. Define one device as the Master Device for Users where User Index ends in 0, 2, 4, 6, and the other for those ending in 1, 3, 5, 7.
3. One single 854 on Channel 6 (Controller No. 4) to be used as an "Auxiliary Device".
4. 512K of ECS with DDP available (Channel 20) for use as an "Alternate System Device".
5. Four 844's configured as two dual-access devices with units 0 and 1 as one device, and units 2 and 3 as the other. They will all be accessed via channels 2 and 3. Make them available to receive a family configured the same as FAMX.

Unit Record Equipment

1. One 405 Card Reader on Channel 11 (Controller No. 4)
2. One 415 Card Punch on Channel 11 (Controller No. 5)
3. One 512 Printer on Channel 22 (Controller No. 6)

Display Console Equipment

One 612 Display Console on Channel 10 (Controller No. 7)

Magnetic Tape Equipment

1. Three 657 Magnetic Tape Drives, Controller No. 5, Units 0 - 2 on Channel 12.
2. Two 659 Magnetic Tape Drives, Controller No. 6, Units 0 - 1 on Channel 13.

Communications Multiplexer Equipment

1. One 6671 Multiplexer, Controller No. 7, on Channel 24 to be used by Export/Import.
2. One 6676 Multiplexer, Controller No. 5, on Channel 23 with 50 lines available.

3. One 6671 Multiplexer, Controller No. 4, on Channel 24 for use by the TELEX time-sharing subsystem. Make 10 lines available.

4. One 6676 Multiplexer entry for use by the Time-sharing Stimulator. Use Controller No. 3, Channel 25, and 10 available lines.

Dummy Equipment

Put TE and NE entries in the CMRDECK even though they are generated automatically.

Miscellaneous Entries

1. Specify a name.

2. Specify 8 Control Points.

3. Increase the length of the File Name Table to 1300_8 words.

28.2 ANSWER SETS

ANSWER TO NO. 1

1) a) 16B (see question 22) from CMR word 1

   b) 27B from CMR word 2

   c) 300000B from CMR word 1

   d) yes from CMR word 1

   e) 35000B from CMR word 20

2) PP3 no IR = *** turned off from CMR word 6230

   PP9 yes IR = 1AJ from CMR word 6310

3) yes PP4 OR=12 RCHM (Reserve channel) from CMR word 6241

   PP11D OR=22 RSYM (request system) from CMR word 6331

   PP12D OR=3 CCHM (check channel) from CMR word 6341

   PP15D OR=12 from CMR word 6371

4) Use EST pointers in CMR word 5.

   EST FWA = 6600, LWA = 6700,

   LWA+1 of MS = 6601 means only one MS device in system

   EQO=DI, EQ10=DS, EQ12=CR, EQ13=CP (off), EQ20=LQ, EQ21=LP, etc. Note EQ30=TT (off). EQ31=TT (for stimulator), EQ40=ST (off).

5) EQ0 for both.

6) Use FNT pointers in CMR word 4

   FNT FWA=6700, LWA=7700 length is 1000B words or 400B files.

   SYSTEM FNT is at CMR word 6700.

   EQ0, track 2 (4002 indicates link to next track in chain), file is rewound since current track = 2 and current sector = 1.

7) a) 6704, SALVARE, EQ0, track 236, rewound, type 7, COMMON

   b) 6714, 1IOVAAN, EQ0, no track, assigned to CP26, type LOCAL=6B.

c) 6726, INPUT*, EQ0, track 306, positioned to sector 2, type input=0, origin=1 assigned to CP2. This is the input file for CP2.

d) 6734, TESTB, EQ0, track 303 positioned to sector 3, type 11=primary terminal assigned to CP4. Note TESTB is an indirect PF so the type is set=11.

8) 47100B from CMR word 23.

9) 10 milliseconds from CMR word 24

10) 16.11.22 from CMR word 30

11) From word 60. IDLE package has the CPU. EXCHANGE address is 17122 which is the address of the IDLE exchange package. CPU assignment is 0, which means no CP has the CPU.

12) Use JBC pointer from CMR word 4

JBC FWA = 10550

JBC order is SYSOT, BCOT, EIOT, TIOT, MTOT

SYOT FWA = 10550

BCOT FWA = 10560

EIOT FWA = 10570

TXOT FWA = 10600

MTOT FWA = 10610

so INPUT QUEUE priority for BATCH is 2400, from CMR word 10560

13) CMR word 10573 CPU priority for EIOT=30.

14) CMR word 10601 ROLLIN queue priority for TXOT = 4004, 3740, and 7000.

15) Use PP resident pointer from CMR word 1.

PP resident FWA = 17646. CIO is at 17646 and is 312 CM words long.

16) Next entry at 20160. PP name is 2CA.

17) a) PLD FWA = 33246 from CMR word 2

RCL FWA = 33245 from CMR word 6

Note that word 33245 = 0 and indicates an empty library.

CLD FWA = 33423 from CMR word 7

b) RPL, CIO
PLD, ADC
RCL, empty
CLD, SYSEDIT

18) a) Dayfile buffer pointers FWA = 6440 from CMR word 3.

Normal dayfile buffer begins at 13520
Account dayfile buffer begins at 10620
Error dayfile buffer begins at 10720
CP1 dayfile buffer begins at 11020
CP2 dayfile buffer begins at 11120

b) 14620 from CMR word 3. Dayfile pointer for CP30 (N+1) = 6524 and has garbage so no dayfile pointer for SYSTEM CP.

19) Channel status table begins at word 100
Channel 0 none from CMR word 100
Channel 3 none from CMR word 100
Channel 10B PP1 from CMR word 101 (PP1 is DSD and CH10 is the display channel.)
Channel 20B PP11 from CMR word 103

20) Channel 17B. Bit 6 is set on. Channel 34B and 35B also are unavailable.

21) CP5 is at address $1\underbrace{01}\underbrace{000}\underbrace{000}\underbrace{0}$ = 1200B

a) AAAY022 type 3 = TXOT. word 1221

b)
```
555 555 555
987 654 321 |#PPs|
000 000 100 |001 |  word 1220
```
W = 0, X = 0, R = 0, job advancement flag is set.

The job is being advanced, it is typed TXOT, so it will be rolled out to TELEX.

c) 1PP is assigned. (It is 1AJ PP9 word 6310.)

d) RA=232600, FL=25600, P=124. from words 1201, 1202, and 1200.

e) CPU = 30, Queue = 7000 from 1222.

f) 102 milliseconds from 1250 or 1254. From 1224 bytes 0 and 1, the time used is 102 ms so job has not run out of time.

g) UN=BOBSIM1, UI=3 from 1264.

h) 2 sectors from 1252.

i) $LDC, O, TESTB,,1. from 1330 and 1331.

j) There are no more control cards.

22) PP3, PP6, PP7, PP13D, PP16D, PP17D. Their IR = ***.

23) The FNT pointers in word 4 are only 12 bits in length and are exact addresses. Hence, 7777B = 4096D is the largest address possible.

ANSWER TO NO. 2

1) CP uses RA+1 calls.
   PP uses OR and IR calls.

2) a) Each PP has one, each CP has one, two idle, and two sub CP exchange packages
   b) All of the above plus disable central exchange return package DXP, and disable central exchange program DXJ

3) a) PP will issue MXN with request in OR.
   b) PP will wait with request in OR for MTR to issue an EXN.

4) a) MTR will issue MXN with request in XO.
   b) MTR will issue EXN with request in XO.

5) a) CP issues XJ with RA+1 request
   b) CP waits for monitor (MTR) to find its RA+1 request

6) a) CPUMTR activates the CP with highest CPU priority that can run in the system at this time.
   b) CPUMTR will insure that the CP which is losing the CPU will have its exchange package in its own CPA.

7) CPUMTR monitor mode is not interruptable. CPUMTR program mode is interruptable (program mode is used for any function which requires more than 250 micro-seconds to complete.

8) a) The CPU will exchange to the address specified in the Bi+K portion of the instruction.
   b) The CPU will exchange to the address specified in the hardware MA register.

9) a) PPR.
   b) PMN
   c) The PP will place the value PPR or PMN in the P portion of its exchange package area

10) a) ART - set up the MSCL parameters for MTR
    b) JAC - check each CP for time slice exceeded
    c) JSW - check each CP active in the CPU for time slot exceeded.
    d) CRC - check each CP in periodic recall for start up, if recall time expired.
    e) PPL - check each CP for a PP routine to start up (RLPW) if PP recall time expired.

11) MTR will drop the queue priority to the lowest queue priority of its origin type and will set bit 35 word TSCW off to indicate time slice no longer active If MTR finds another CP with a higher or equal CPU priority it will start up CPUMTR.

12) It will issue a recall CPU monitor function RCLM, which will call CPUMTR to set recall bit off and W bit on.

13) CPUMTR.

14) CPU priority and CPU time slot exceeded then call CPUMTR which will determine which CP to give the CPU.

15) The highest CPU priority If several CPs have the same CPU priority, then they get the CPU in rotation.

16) Need two pseudo CPA with at least 20 words for the exchange package. Assemble two routines and absolutize relative to RA+X, where X is the length of the communication area defined. Use LDR to load these two routines and use XJP RA+1 requests.

ANSWER TO NO. 3

1) PPFW It must load at PPFW-5 for the 5 byte header, and if using MS drivers should not use FL beyond 7000B.

2) To insure proper running of the system it is necessary to move CP field length in CM. A CP cannot be moved while it has PP activity Hence, a PP should pause occasionally for this movement.

3) It is looping on reading its IR.

4) a) It uses the routine FTN which will place the request in its OR and if possible or necessary it issues an MXN.
   b) The request has been honored when the B0 in the exchange package has gone to zero.
   c) The request has been completed by monitor when the PPs OR has been set zero

5) Channels are interlocked by CPUMTR in the channel reservation table in CMR.

6) The CP assignment is in its IR byte 2, and PP res will store it in direct cell CP

7) It uses the DFM routine of PP res

8) In the direct cells RA and FL, which are set by the PP res routine PRL, which gets these values from the STSW word of CP area

9) It uses the EXR or PLL routine in PP res. Set (A) = routine name, (LA) = load address for location free routines.

10) Set (T5) = est ord and RJM SMS.

11) POS, WDS, and RDS.

12) These locations are used for buffers and error processing.

13) Left to the reader.

ANSWER TO NO. 4

1) Whenever there is a change in system resources available or time slices or CM time slices expire or periodically

2) RSJM.

3) EPR via AFP checks JCB for interval and integer to add to each queue value in the Input and rollin queues.

4) a) MTR checks for time slice exceeded.
   b) MTR does an RSJM. 1SJ in routine ERP via AJP will drop the queue priority to lowest queue priority for this origin type.

5) Yes, Yes, Yes

6) If it finds a new job with a higher queue priority which needs the rollout of a CP to get FL.

7) Highest queue priority which can get FL, after all other jobs scheduled or scheduable for rollout has released their FL. If a tie then job with largest FL requirements is chosen.

8) a) 1AJ to start a job.
   b) 1RI to roll in a job.

9) 1SJ calls it into his own PP after issuing an RSJM.

10) Zero status (ie no activity) on a CP or rollout flag set

11) 3AA, 3AB, and TCS.

12) 3AA is called when 1AJ wishes to begin a job. (i.e 1SJ called 1AJ). 3AA will read the control cards into the CPA and process the 1st card, processes the job card and initiates job processing.

13) No, TELEX type jobs are always in rollin status

14) When an error flag is set and 1AJ prepares to advance the job. 3AB will look for an EXIT card and if it finds one will continue advancement with the card following the EXIT card.

15) TCS

16) CTIME, RTIME.

17) 1AJ via TCS cracks a control card and uses LDR (or LINK) to load the proper routine and request CPUMTR to set its W=1 If it is a PP routine call see question 21.

18) It checks the Local FNT entries and when 1AJ finds a match it calls LDR to load it.

19) It calls LDR

20) He cracks it and processes it like any other control card (i.e., a call to LDR or PP load).

21) Yes, TCS uses routine CPP which loads PP into this PP over itself after finding it in the RPL or PLD.

22) See Chapter 23 program BOB.

23) If you use the job in Chapter 23 with the Control cards following.
JOB
ACCOUNT
COMPASS
SYSEDIT
BOB
DMP (fl)
then the SYSEDIT code will be in FL when the card BOB is encountered. Since this routine has a DMP= SEP, 1RO will create a DM* file with SYSEDIT on it BOB will run, and when it completes, 1RI will copy the SYSEDIT code from the DM* file back into the FL. The DMP will dump the SYSEDIT field length.

24) DMP has a DMP= SEP so 1RO will create a DM* file with the SYSEDIT fl and DMP will dump the SYSEDIT FL and end. 1RI will copy the SYSEDIT code back into the FL, but there are no more control cards to process so 1CJ will complete the job.

25) SSCL and SSCL+1

26) It will create an input queue entry for the subsystem (i.e. an FNT/FST entry type input).

27) a) It is the collection of FNT/FST entries whose type are PRFT and which are assigned to CP0.

b) same as (a) except type is ROFT.

28) They differ primarily in the FNT entry which is a) jobname for rollout file b) DM* for DMP= SEP job The actual DM* file on MS is the same as the rollout file if a full rollout dump was requested by the DMP= SEP job

29) The ACCOUNT card cause the routine ACCFAM to be loaded and it has a VAL= SEP

ANSWER TO NO: 5

1) SMS loads the proper disk driver into PP resident at location MSD (MSD=600).

2) FET→FNT → EST → MST/TRT.

3) A sector count is specified when requesting mass storage space so that a track chain can be established. (Allocation and deallocation of track chains in done by CPUMTR) The number of sectors/track in contained in the MST for any particular device.

4) RDS - read sector
WDS - write sector
POS - position disk

5) The drop track monitor function (DTKM) is issued after writing a disk file so that any unused tracks will be returned to the pool of allocatable tracks. Additionally, the EOI sector number is stored in the TRT.

6) File STIMFL FNT/FST is at location 6720 and 6721. It is type 6=LOCAL, assigned to CP 27B, ID=0, 1st track = 243, current track - 243, current sector = 3, status is 15 = 001 101 means file not busy, open for read, last operation read FNT-EQO→EST-6600→MST-7700: TRACK 243 is $\boxed{010\ 100\ 0}\boxed{11}$ = TRT word 50 byte 3 TRT is MST+20 = 7720 - word 50 = 7770 byte 3. The track is 3 sectors long, last sector is EOI So file positioned at EOI. (Note: this file is being used by the STIMULATOR)

ANSWER TO NO. 6

1) For a write request, an FNT entry will be created For a read request, EOI status is returned in addition to creating an FNT entry.

2) a) No other PP routine is called by CIO because all mass storage I/O and positioning routines are contained in overlays called by CIO.

   b) CIO extracts the function code from the FET, loads an appropriate overlay and performs the requested read/write until the operation is complete

3) Random I/O is accomplished by the user specifying logical addresses of records whereby CIO converts the logical address into a logical disk position of the particular record. A random read operation is performed as follows:

   1. User sets FET+6 = logical address of the record desired and issues a random read request (in FET+0)
   2 CIO converts the logical address to a random index, positions the disk and returns the data.

4) A re-write in place stores a new record in the same sectors occuppied by an old record. An EOI sector is not written for any random write request. A sequential write sets EOI.

5) CIO stops on detecting EOR, EOF or EOI status. CIO will drop if the buffer is full, or if FL/100B sectors have been stored in the buffer.

6) CIO sends a 3-word parameter block to a specific UDT within MAGNET's FL.

7) CIO stores the FET address in the control point area word TIOW and TINW and issues monitor function ROCM to roll out the job

8) Lots. Very careful and extensive modifications are needed.

ANSWER TO NO. 7

1) Control cards processed by RESEX include:

    LABEL
    ASSIGN
    REQUEST
    VSN
    RESOURC

2) RESEX is a special entry point program with the DMP= entry point. Therefore, the user's job is rolled out to the DM* file prior to loading RESEX After RESEX completes, the user's job is rolled back in from the DM* file.

3) The PREVIEW display informs the operator of outstanding magnetic tape and removable pack requests.

4) RESEX updates the PREVIEW display.

5) UDT entries are stored in MAGNET's field length.

6) The RESOURC control card declares the maximum number of concurrent tapes and packs to be used by a job. Depending on the availability of such resources, the job will be allowed to continue or will be suspended temporarily.

ANSWER TO NO. 8

1) According to the CMR dump, the first track of the Indirect Access file chain is track 234 (see byte zero of MST+4 word 7704). Permit buffers start on track 235 (see byte 2 of MST+4). EST-EQO at 6600→MST at 7700.

2) According to the CMR dump, there is one track allocated to indirect access permanent files Pointer is in MST+4 (7704) track is 234 which is word 47 byte 0. There are 12 sectors and no track links. The total for each user can be calculated by adding individual file sizes as stored in the catalog entries.

3) a) One track is allocated for permit buffers. 235 = $\boxed{010\ 011\ 1|01}$ 7720+47=7767
4 7 ∠byte 1

b) 20 tracks for catalog tracks. All from word 7704

4) Yes. Three direct access permanent files exist

5) The user index is set to zero in the catalog, all complete tracks are returned to system and the TRT entries are relinked if necessary. If tracks are released then the catalog entry is adjusted accordingly.

6) Holes in the Indirect file chain are not pointed to PFM searchs the holes for an exact fit (i.e. same number of sectors needed) and if one exists it is used. If no exact fits, then PFM uses the largest hole and creates a new catalog entry with UI=0 for the residue. If no holes will satisfy the request, then a new catalog entry is created for the file.

7) USER*** will have read-only permission for file XYZ. (See page 2-6 of K2.1 Reference Manual).

8) Either a semi-private file or a library (public) file can be accessed by any user by specifying the permanent file name, the user number under which it was created and the password if defined. The system records the number of times the file was accessed for either file category. However, for the semi-private file the user numbers and last access date and time are also recorded (ref PF control cards in reference manual).

9) Yes. Indirect access files must reside on the user's MASTER device. Direct access files need not reside on the user's MASTER

10) Yes. Each job has an FNT pointing to the same file.

11) With indirect access permanent files the user is given a local copy of the file.

12) Attaching a direct access permanent file with "write" permission essentially locks the file in that no other user can access it until it is returned This prevents two users from modifying the file at the same time. However, if any users have attached the file in read mode prior to the attach with write, the file cannot be written on until all read only users have released the file.

13) When a direct access permanent file is purged, its track chain is returned to the system and its catalog entry is released.

14) Abort. Yes. To avoid this situation, DEFINE the file prior to writing it.

ANSWER TO NO. 9

1) OVL has 1 entry point.
   ABS has multiple entry points.

2) Most utility programs need multiple entry points for different functions.

3) 77 ident table.

4) a) LINK b) It is loaded by LDR.

5) a) It will call LDR. (Such as a LOAD an ABS binary file)
   b) It will process it.

6) LDSET (optional) used for the names of all Libraries to use for satisfying externals.
   PIDL used for program name and length

7) LDSET, LOAD, MAP, NOMAP, LIBRARY, REDUCE, and SETCORE.

8) a) They are used to satisfy externals. Also a Library can be specified for externals in place of the normal Library for the same externals (i.e. different external routines can be linked by the loader other than the standard one. e.g. instead of using the FORTRAN library use a user defined library).

9) Yes, LDSET card

10) LDR. LDR uses the 2 or 4 word request created by the macro to load the routine in the FL.

11) a) High usage routines can be put on a separate device for quicker access b) Use CMR DECK entry ASR = E ord. Use the SYSEDIT directive * AD as shown in section 11.3 ASR c) There is no operator command Either change PLD pointer in CMR core from the console, or disable the alternate device.

12) Because different PSR level Libraries may make non-compatible calls to each others routines.

ANSWER TO NO. 10

1) Programs comprising the time-sharing subsystem include:
   TELEX/TELEX1/TELEX2 - CPU routines 1TA, 1TO and 1TD - PP routines

2) The major elements of the TELEX control point are:
   - buffers for data to/from TTYs (POTs)
   - queues
   - command processing routines
   - queue processing routines

3) The dynamic memory associated with TELEX is used for POTs. All POT allocation and deallocation is controlled by TELEX via a POT Link Table (PLT) which is similar to a TRT.

4) TELEX job flow -
   a) Job initiaition -
      1. TTY operator enters a command.
      2. TELEX calls 1TA to set the user's rollout file to a status whereby it is a candidate for rollin.
      3. 1SJ selects the job to be rolled in.
      4 1RI rolls the job into a control point
      5 1AJ read the next control card and loads the appropriate routine.
   b) Terminal input -
      1. time sharing job issues a read on the INPUT file
      2 CIO issues monitor function ROCM
      3. 1RO rolls out the control point and calls 1TO.
      4. 1TO sotres any data in POTs and informs TELEX of the request for input
      5 TELEX issues "?" to TTY
      6. TTY user enters data.
      7. TELEX calls 1TA to create a rollin queue entry
      8. 1SJ selects this entry
      9. 1RI rolls the job in and transfers data from POTs to the circular buffer.
   c) Terminal output -
      1. The time sharing job issues a write on the OUTPUT file.
      2 CIO calls 1RO to rollout the control point.
      3. 1RO calls 1TO to store data in POTs.
      4. TELEX sends data to TTY.
      5. TELEX calls 1TA to create rollin queue entry
      6. 1SJ selects this entry
      7. 1RI rolls the job into a control point.

5) Requests from the driver (1TD) to TELEX are placed in the driver request queue Requests from TELEX to the driver are placed in byte 4 of the VDCT word of a terminal table entry

6) PP programs 1TA and 1TO are called by TELEX via a TLX request in RA+1 (parameters are stored in POTs). These PP routines issue monitor functions TGPM and TSEM to request TELEX processing.

7) The reentry table enables TELEX to return control to a subroutine which was suspended until a set of conditions were met.

8) TELEX internal queues are:

    Monitor Request Queues - process TGPM and TSEM requests
    WCMQ - wait for completion of a process
    TIMQ - time delay queue, wait for time to elapse
    JOBQ - perform all job scheduling at one time
    SORQ - perform scheduling of sort jobs
    PP queues - send all 1TA, 1TO and PFM requests
    Driver Request Queue - process driver requests

9) Information concerning a particular terminal is kept in a terminal table entry within TELEX. There is one entry for each known port and each entry is eight words in length.

10) A multi-terminal job is a job that performs operation for many terminals and is loaded only once. (Currently used for sorting several primary files).

11) Terminal requests (i.e., driver requests) are processed twice in the main loop of TELEX via subroutine DR1.

12) CPUMTR informs TELEX that the PP request cannot be honored by not clearing the PP call word pointed to in the TLX RA+1 request. TELEX increments a count of such occurrances and tries later. There is no special case processing done by CPUMTR since all TLX requests are handled the same.

13) The time-sharing subsystem is activated by the operator command: TELEX This causes DSD to call 1DS. 1DS in turn builds an input FNT entry with the name 1TD specified. 1SJ then calls 1TD to the specified control point (in this case, control point 1). 1TD then performs control point initialization and builds a control card buffer in the control point area with the following cards:

    TELEX
    TELEX2.
    EXIT.
    TELEX2

13) cont

1TD then calls 1AJ to process the first control card TELEX TELEX initializes tables and pointers and loads the main routine TELEX1.

14) 1TA requests are grouped together by TELEX to minimize the number of PP calls required.

15) TELEX jobs are scheduled to the rollin queue instead of the input queue since the rollout file is available and in fact is created during log-in processing.

16) So that jobs can be scheduled to the rollin queue instead of the input queue.

17) The job is rolled out, any output is sent to the terminal and the FNT entry for the rollout file is removed from the system FNT. TELEX regains control and has the FST entry for the rollout file.

18) 1TO performs terminal I/O for TELEX as CIO does for batch jobs. However, 1TO transfers data between POTs and an RMS device.

19) 1TD checks for input after each output character is sent to check for an interrupt request from the TTY operator.

20) The reentry address is stored in byte 4 of terminal table word VDPT. This is the address within the driver where control is passed during the next pass.

ANSWER TO NO. 11.

1) a) It needs a CP routine for data conversion, and must have a subsystem CP for overall communication between the PP routines.

   b) It has queue priority greater than MXPS.

   c) It has a very high CPU priority, however, TELEX and the system CP have a higher CPU priority and can get the CPU from E200CP.

2) Function word, status word, message buffer, LINF, CP1K, DPJT or PWLT, FAMT, and input and output FETs and buffers.

3) Via the Function word table.

4) a) 1LS

   b) a PP must set up the subsystem status, CPA and Fl.

   c) 1DS builds the input queue entry for 1LS. 1SJ will start 1LS with a CP assigned, and 1LS will build the CPA and initialize the FL.

5) While 1ED is transmitting print data or receiving card reader input to the 200 UT, it will monitor the keyboard. All entries from the keyboard are ignored except the interrupt key If 1ED senses the interrupt key, it will stop transmitting and receive input from the keyboard. At that time the 200 UT operator can type END, CR or LP

6) XSP will a) process job drop requests, b) Log in terminal processes, c) make initial input queue entry.

7) CIO.

8) See overview diagram Step 1, 2, 3, 4, and 5.

9) See overview diagram Step 6, 7, 8, and 9.

10) Compares Jobname from FNT with L1NF table.

ANSWER TO NO 12.

1) 1IO, 1CD, 1BA.

2) For PP communication, FETs and buffers.

3) It has queue priority greater than MXPS.

4) 1IO is the only PP and it is in PP recall via the RLPW word in the CPA

5) It calls CIO.

6) 1IO calls 1CD calls 1BA to create an input queue entry (i.e. an FNT/FST entry of type INFT).

7) The LPxx, yy or LQxx, yy command will place yy in the unit field of the EST at ordinal xx if it is an LP or LQ type Then BATCHIO will compare the ID field (if not zero) of the output queue entries to the LP and LQ devices until it finds a match in the unit fields.

8) They are used by 1IO to determine how many function each copy of 1CD is processing

9) a) The DRQR is used to request a function from 1CD
   b) 1CD compares its DCW offset in its IR against the DCW offset in the DRQR

10) When the p bit of its IR is zero.

ANSWER TO NO. 13.

1) So that they can input all of their FL before being released from the IAM instruction.

2) They are hung on an IAM channel, 0; where channel number corresponds to PP number

3) PRL is coded to expect them in that order.

4) It is read into PP0 word 1 thru 14.

5) It builds the PPULIB, RPL, RCL, PLD and CLD.

6) It contains all the PP routines not in the RPL. The PP routines have their 77 table stripped from them, but the complete PP routines still reside on the SYSTEM file The PLD points into the PPULIB file.

7) a) STL issues an RSJM monitor function
   b) to start scheduling of the system.

8) KRONOS always recovers, the different levels only impede the process in some way.

ANSWER TO NO. 14.

1) When it must perform a function for a particular CP. Such as enter data into CM, but does not include overlays displays for a particular CP such as K, n.

2) When DSD must perform some function it cannot do it calls 1DS to perform it.

3) No, they build one line of display at a time; and the main loop displays that line.

4) DSD senses X. DIS and finds an available CP. DSD calls 1DS which calls DIS into its PP with the CP assignment made by DSD

5) They sense "*" from the keyboard. If "*" is sensed as the 1st character, they will release the channel. When not connected to the channel they periodically check the EST ordinal for device DS and when it is available (the CP assignment is 0) they will request the channel via RCHM monitor function.

ANSWER TO NO. 15.

```
CMRDCK1
NAME = CDI KRONOS 2.1 TIME-SHARING SYSTEM
NCP = 10.
FNT = 1300.
EQ0  = DB, ON, 1, 0, 0
EQ1  = DB, ON, 1, 1, 1.
EQ2  = DI-2, ON, 0, 0, 2, 3, R
EQ3  = DI-2, ON, 0, 2, 2, 3, R.
EQ4  = DI-2, ON, 0, 0, 4, 5, R.
EQ5  = DI-2, ON, 0, 2, 4, 5, R.
EQ6  = DD, ON, 4, 0, 6.
EQ7  = DP, ON, 2000, 20.
TEMP = 0, 1.
FAMILY = 2.
PF = 2, D, 125, FAMX, 40
PF = 3, D, 252, FAMX, 41.
REMOVE, 4, 5, 6.
ASR = 7.
EQ10 = DS, ON, 7, 0, 10
EQ11 = CR, ON, 3, 0, 11.
EQ12 = CP, ON, 5, 0, 11.
EQ20 = LP, ON, 6, 0, 22.
EQ30 = ST, ON, 7, 0, 24.  (6671 for EXPORT)
EQ40 = TT, ON, 5, 0, 23, 50.
EQ41 = TT, ON, 4, 0, 26, 0, 10.  (6671 for TELEX)
EQ50 = MT-3, ON, 5, 0, 12.
EQ60 = NT-2, OFF, 6, 0, 13.
EQ76 = TE, ON, , , .
EQ77 = NE, ON, , , .
```

APPENDIX A
GLOSSARY

ABS	Absolute
ABT	Abort
AC	Account Number
ACP	Advance Control Point
AL	Alternate Library
ARB	Accounting Record Block
ARBS	Accounting Record Block Size
ARS	Alternate System Residency
ART	Advance Running Time
ASR	Alternate System Residency
ATX	After Exchange Jump
AVC	Advance Clock
AUN	Alternate User Number
B	Binary Requested
B	Octal Number
BCE	Begin Control Point Executive
CC	Command Code
CBP	Check Buffer Parameters
CCE	Create Catalog Entry
CCP	Check Central Program
CEJ	Central Exchange Jump
CFA	Check File Access
CH	Channel
CIO	Combined Input/Output
CLD	Central Library Directory
CM	Central Memory
CMR	Central Memory Resident
CMS	Copy Mass Storage
CP	Control Point
CPA	Control Point Area
CPE	Control Point Executive (Also E)
CPR	CPUMTR Request Processor
CPS	Control Point Status
CPU	Central Processor Unit
CPUMTR	Central Processor Unit Monitor
CRC	Check CPU Periodic Recall Status
CRR	Check Rollout Request
CSF	Check Special Format
CUC	Check User Controls
D	Decimal Number
DDP	Distributive Data Path (ECS)
DFD	Dayfile Dump
DFM	Dayfile Message
DT	Device Type
E	Executive (Also CPE)
ECP	End Central Program
ECS	Extended Core Storage
EF	Error Flag
EFN	Enter File Name
EM	Error Modes
EOF	End of File
EOI	End of Information
EOR	End of Record
EP	Exchange Package
EPA	Exchange Package Area
EPT	Entry Point
EQ	Equipment
EST	Equipment Status Table
FE	Fatal Error
FET	File Environment Table
FL	Field Length
FM	File Managers
FNT	File Name Table
FP	First Pot
FST	File Status Table
FWA	First Word Address
HNG	Hang
ID	Identification
IDL	Idle Loop
IIQ	Initial Input Queue
INT	Initialize
IOQ	Initial Output Queue
IQ	Input Queue
IQP	Initial Queue Priority
IR	Input Register
IRQ	Initial Rollout Queue
ISR	Identify Special Request
JAC	Job Activity
JCA	Job Control Area
JSW	Job Switching
KTS	KRONOS Transaction Subsystem

GLOSSARY (Continued)

LCP	Load Central Program
LDB	Load CM Buffer
LFM	Local File Manager
LFS	Location-Free Subroutines
LIQ	Lower Input Queue
LOQ	Lower Output Queue
LP	Last Pot
LQP	Lowest Queue Priority
LRQ	Lower Rollout Queue
LWA	Last Word Address
MA	Monitor Address
MAGNET	Magnetic Tape Executive
MB	Message Buffer
MEJ	Monitor Exchange Jump
MF	Monitor Flag
MM	Monitor Mode
MS	Mass Storage
MSD	Mass Storage Driver
MSG	Message
MST	Message Storage Table
MTR	Monitor Peripheral Processor
NFN	New File Name
OQ	Output Queue
OR	Output Register
OVL	Overlay
PCW	Program Control Word
PF	Permanent Files
PFM	Permanent File Manager
PFN	Permanent File Name
PGM	Program
PLD	Peripheral Library Directory
PLL	Peripheral Library Loader
PLT	Pot Link Table
PMS	Position Mass Storage
POS	Position Disk
PP	Peripheral Processor (Also PPU)
PPC	PP Communication Area
PPR	PP Resident
PPU	Peripheral Processor Unit (Also PP)
PRU	Physical Record Unit
PTX	Prior to Exchange Jump
Q	Queue
QP	Queue Priority
RA	Random Address
RESEX	Resource Executive
RCH	Reserve Channel
RCL	Resident Central Library
RDS	Read Sector
RI	Rollin
RJC	Read Job Control
RMS	Rotating Mass Storage
RO	Rollout
RPL	Resident Peripheral Library
RTS	Request Task List
SAF	Search for Assigned Files
SCP	Sub Control Point
SED	Set Equipment Definition
SEP	Special Entry Point
SFJ	Search for Job
SFP	Special Function Processor
SFS	Set File Status
SIB	Search Index Block
SL	System Library
SLM	Sector Limits
SLP	Set Load Parameters
SLT	Search Library Table
SMS	Set Mass Storage
SN	Sector Number
SRP	Special Request Processor
SS	Sub System
SSF	Search for System File
SSP	Set Statistical Parameters
SSS	Sub System Subroutine
SUN	Search for User Number
TCS	Translate Control Statement
TELEX	Time-Sharing Executive
TFS	Tree File Structure
TRT	Track Reservation Table
TRANEX	Transaction Executive
TI	Time In
TIO	Terminal Input/Output
TO	Time Off
TLD	Task Library Directory
TN	Terminal Number
TT	Terminal Type
UDT	Unit Descriptor Table
UFS	Update File Status
UI	User Index
UIQ	Upper Input Queue
UOQ	Upper Output Queue
UQP	Upper Queue Priority

GLOSSARY (Continued)

URQ	Upper Rollout Queue
VUN	Verify User Number
WC	Word Count
WCB	Write Central Buffer
XF	Execute Flag

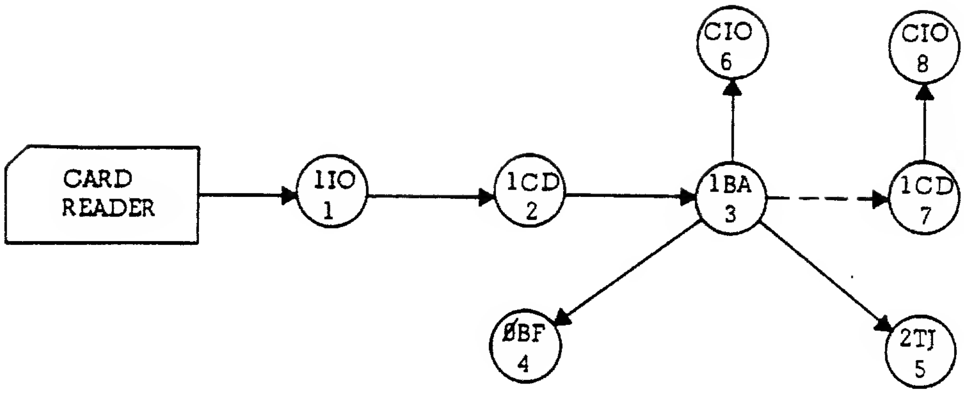
APPENDIX B
KRONOS 2.1 GENERAL OVERVIEW

LEGEND

1. Solid Horizontal line ——→ direct call
2. Dashed Horizontal line – – → return of control
3. Solid vertical line upward ↑ external of helper routine
4. Solid vertical line downward ↓ overlay
5. Downward double line ⇓ routine completely overlays caller
6. Circle is PP routine
7. Square is CP routine
8. Ø is zero
   O is alphabetical O
9. UI is user index
10. CP is control point; CPA is control point area
11. Solid double vertical line double headed ⇕ routine calls overlayed routine back
12. FL is field length
13. MS is mass storage
14. EST is equipment status table
15. EF is error flag
16. XXX is PP routine name, XXXXXXX is CP routine name
17. SEP - special entry points
18. PBA field in SPCW of CPA indicates 20 word parameter block

A. INPUT

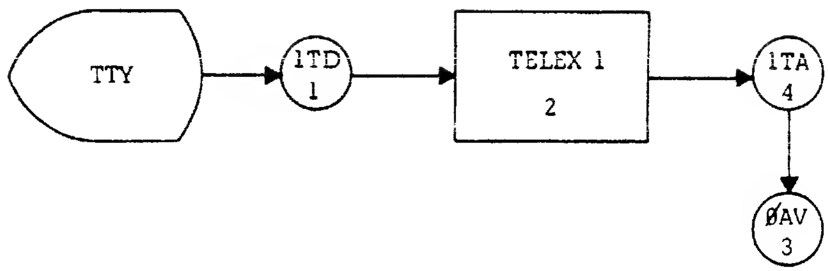
I. BATCHIO (BCOT)



1. RLPW in CPA activates executive
2. DRQR request queue activates driver which reads one sector from card reader to CM buffer.
3. Driver calls 1BA to initiate job
4. 0BF creates FNT/FST entry
5. 2TJ cracks job card and
   a. writes job card information
   b. generates job name + RJSM
   c. creates system sector
6. Copy first sector from CM buffer to input MS file
7. Driver copies rest of cards to CM buffer
8. CIO copies from CM buffer to INPUT MS file

A. INPUT (Continued)

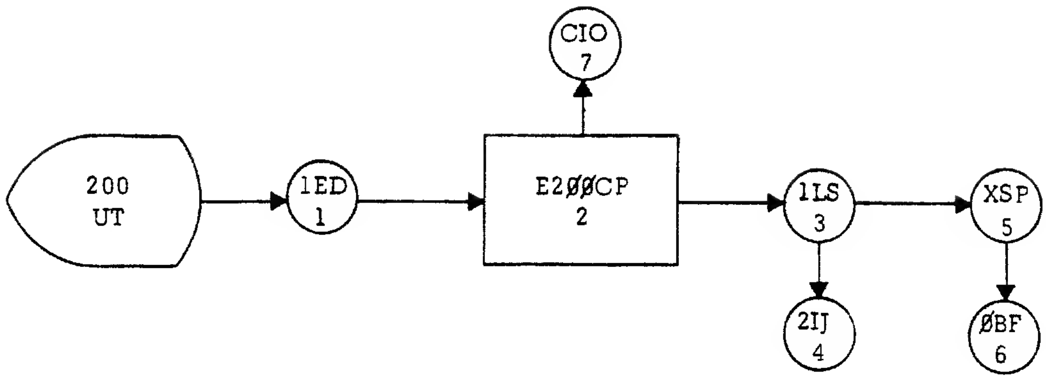
II. TELEX (TXOT)



1. Driver transfers login data to TELEX
2. TELEX 1 sets up ITAQ for jobs destined for a CP.
3. Validate user and set UI. (1TA log in function)
4. ITA builds a ROLLIN/ROLLOUT queue entry for user
   a. generates job name via COMPGJN
   b. creates system sector
   c. creates FNT/FST entry in terminal table
   d. all local files will be carried in the ROLLOUT file for 1RI to create
   e. create FNT/FST entry for above (when job is destined for a C.P.) (Does not use OBF)

A. INPUT (Continued)

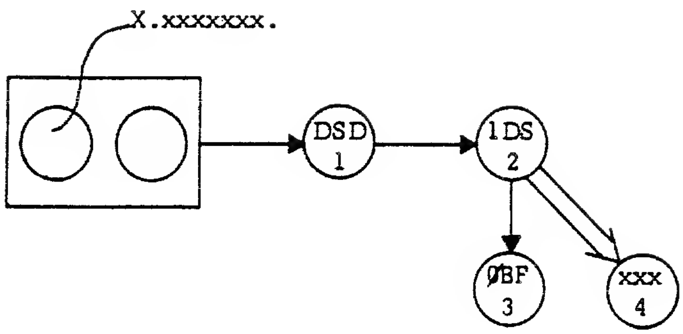
III. EXPORT/IMPORT (EIOT)



1. Driver receives activity from remote card reader and starts CP routine.
2. CP routine starts executive 1LS and transfers data from 1ED buffer to CIO buffer.
3. Executive calls job card processor 2TJ to put job priorities into FET+7, generate job name and calls input file processor XSP.
4. Same as BCOT step 5 except append RJSW to 1LS generated job name.
5. Generate system sector
6. Create FNT/FST entry
7. Copy rest of input data to MS file.

A. INPUT (Continued)

IV. CONSOLE (SYOT)

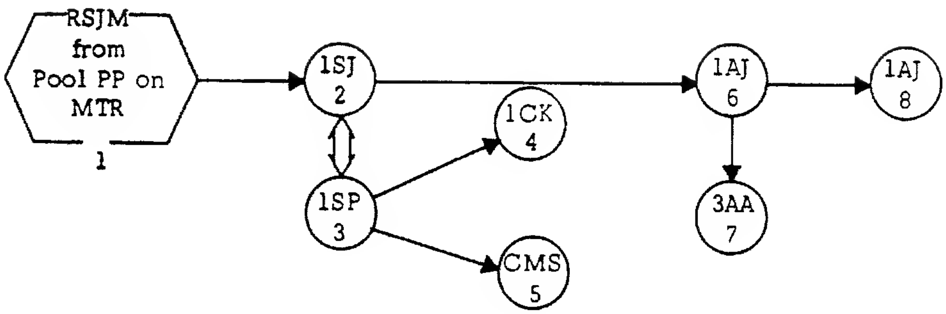


1. Process operator command
   a. if control card call, call 1DS function 17-1CJ
   b. if subsystem call, call 1DS function 17-1CJ
2. a. 1. if PP call see 4
      2. generate job name and start CP job
   b. generate subsystem input queue entry and system sector
3. Generate FNT/FST entry
4. Call PP requested directly

B. JOB FLOW

I. SCHEDULER (1SJ)

1. Start Up.



1. 1SJ is started only by CPUMTR in response to RSJM and controlled by CMR words JSCL and JSCL+1.

2. Locate job in INPUT or ROLLOUT queue with largest queue priority
   a. get CP
   b. get FL

3. 1SP is periodically called to check on MS and age priorities.

4. Update MST/TRT for MS specified by checkpoint requested bit in EST

5. Update removable MS and process INITIALIZE requests

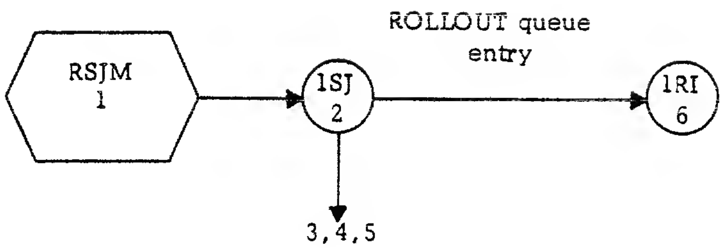
6. a. Initialize CC buffer (position to second card i.e., ignore job card)
   b. Set VAL = bit 17 in UIDW if ACCOUNT/VALIDATION enabled from CMR word SSTL

7. Initialize CPA.

8. Go to VALIDATION or NORMAL processing depending on ACCOUNT/VALIDATION enabled.

B. JOB FLOW (Continued)

2. Continuation



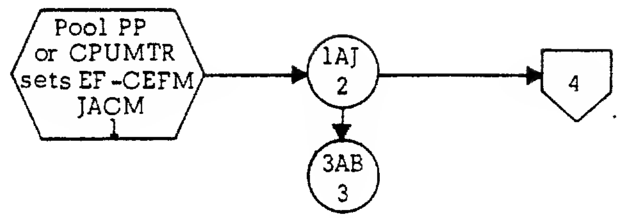
1,2,3,4,5 same as start up

6. Roll job into CP.

B. JOB FLOW (Continued)

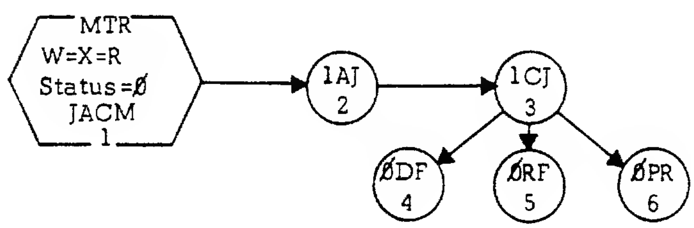
II. ADVANCEMENT

1. Job Error



1. JACM sets job advancement flag and calls 1AJ.
2. Check EECW and STSW (error flags) in CPA.
3. Analyze error and issue error message.
4. a. If reprieve or user control on errors, return to user
   b. EXIT card processing

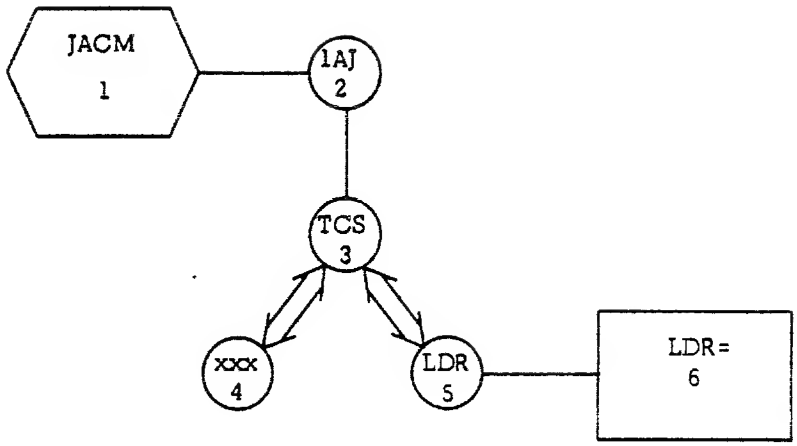
2. Termination



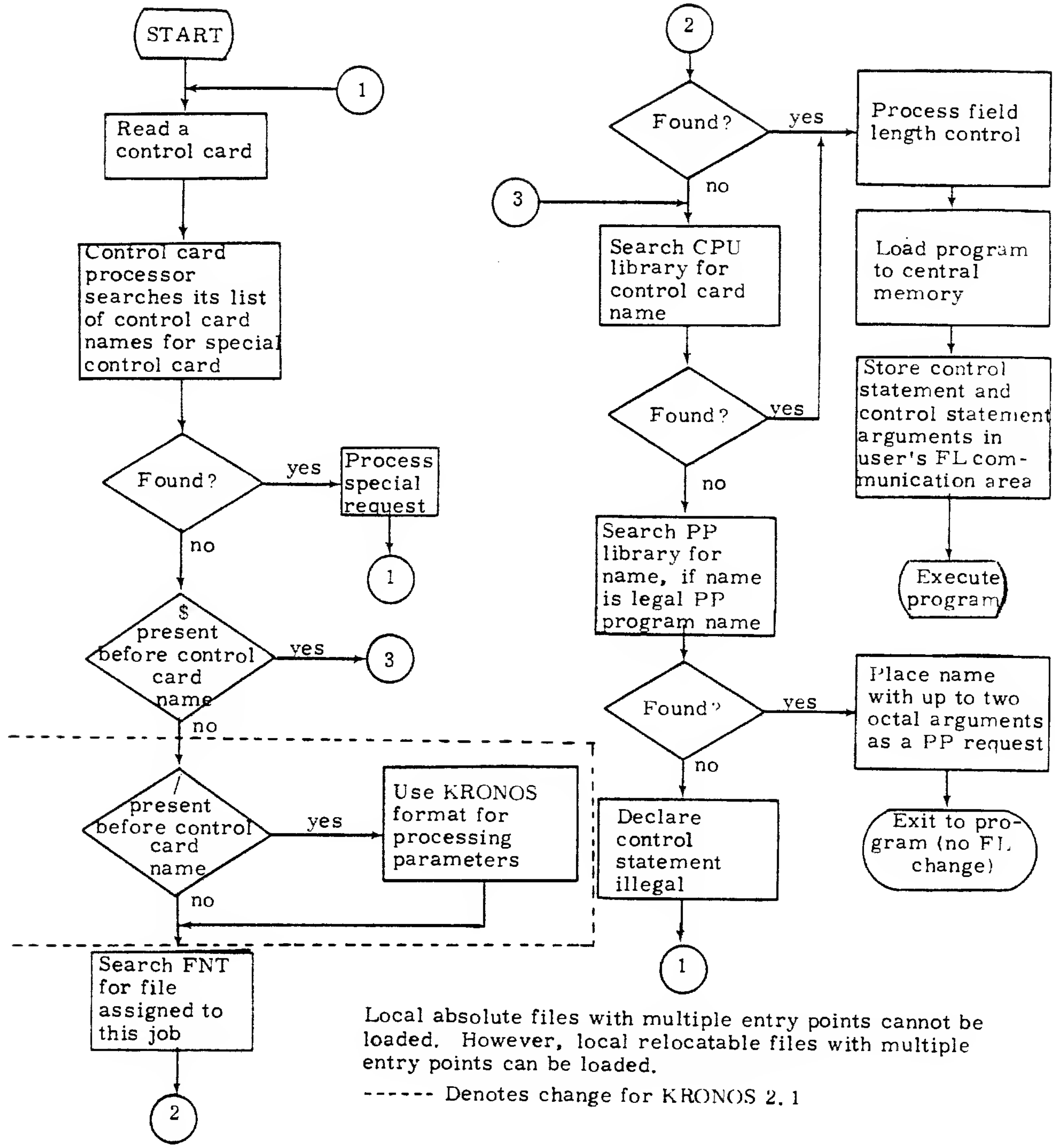
1. Last CC completed and MTR finds advancement status = 0
2. 1AJ discovers EOR on CC statement file
3. Complete job
   a. dispose OUTPUT files, append dayfile and change SYOT to BCOT.
   b. release CP and FL
   c. Complete job accounting UADM
4. Drop all local files
5. Update resource files (RESEXDF and RESEXVF)
6. Release permanent files.

B. JOB FLOW (Continued)

3. Normal



1. Same as termination step 1.
2. 1AJ verified that CC statement buffer is not at EOR.
3. TCS
   a. processes next CC
   b. locate RPL, PLD, CLD entry or $LDC call. In fact 1AJ never explicitly searches the RCL it finds all RCL entries in the CLD.
   c. Crack arguments if no ARG=
   d. CC dayfile message if no SDM=
4. If a PP call, load PP routine
5. Process ABS CP program load (see next page) process RFL=OR MFL=if present.
6. Process REL CP program load (see next page).



Local absolute files with multiple entry points cannot be loaded. However, local relocatable files with multiple entry points can be loaded.

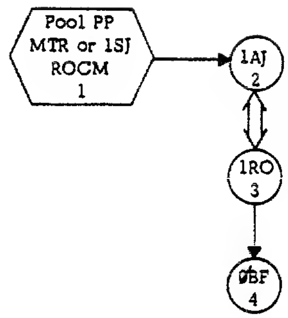
------ Denotes change for KRONOS 2.1

*Load routine only if it is ABS/OVL, else declare illegal cc.

Control Card Processing

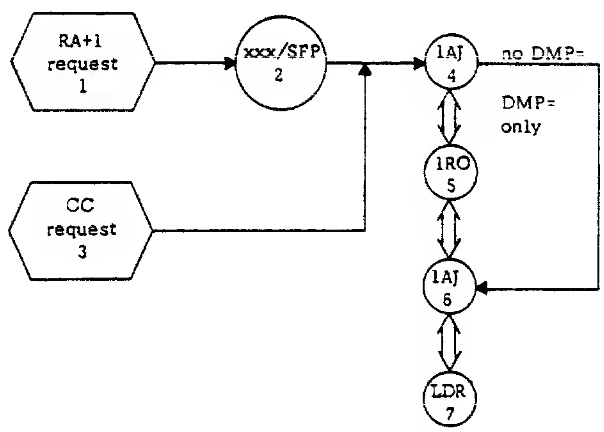
B. JOB FLOW (Continued)

4. Rollout



1. ROCM sets the rollout bit STSW in CPA
2. 1AJ finds rollout bit set
3. 1RO can only be called by 1AJ
   a. Create rollout file
   b. Create system sector
4. Create special FNT/FST entry (if needed (DM*)

5. Special Entry Points (DMP=, SSJ=)



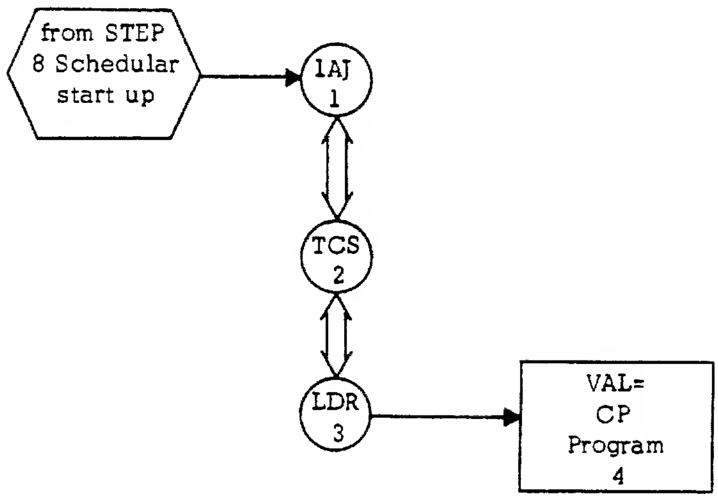
1. Cause loading of xxx or default loading of SFP.
2. Set up SPCW work in CPA
3. Normal CC processing (notice it's an SEP)
4. Set up SEPW word in CPA from SEPA word in CLD
5. Set up DM* file according to DA field in SEPW & process PBA if non-zero. PBA is only available with DMP=SEP.
6. If SSJ= set up SSJ = CPA areas from SA field in SEPW
7. Load ABS CP program

C. VALIDATION (VAL=)

I. TXOT - See INPUT.

II. BCOT/EIOT

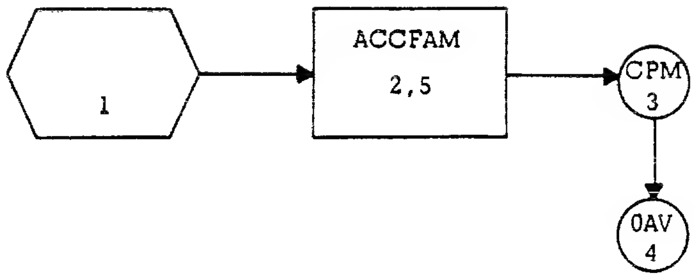
Step 1. Begin job time



1. 1AJ verified that CC statement buffer not at EOR
2. Process next CC - see step 3 of NORMAL ADVANCEMENT.
3. Verify that CC program call has VAL=SEP in SEPA of CLD
   a. load program if VAL=
   b. set EF if no VAL=, abort CP and issue error message
4. VAL= CP program loaded must be ACCFAM (called by ACCOUNT CC) or CHARGE (called by CHARGE CC) since no other routines exist in KRONOS 2.1 as of June 1974.

C. VALIDATION (VAL=) (Continued)

Step 2.



1. CHARGE if loaded will abort CP since UI=0 in CPA.

2. ACCFAM has an SSJ= SEP so it can use UI= 377777B and attach VALIDUX file.

3. CPM in VALID mode will set UI from 0AV.

4. 0AV will validate user on VALIDUX file.

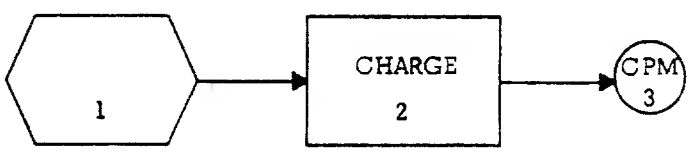
   UI= 0 if not verified
   value if verified

5. Place Validation parameters in CPA or abort user with message if UI=0
   a. set VAL= flag in UIDW off if CCNR in AACW in CPA indicates CHARGE card unnecessary, via SSJ= param block
   b. subsequent ACCFAM calls will only reset UIDW word, via SSJ= param block, all other verification areas in CPA remain unchanged.

Step 3.

Ignore if CCNR in AACW on, indicating CHARGE processing unnecessary.
Else same as Step 1.

Step 4.



1. If ACCFAM loaded again see Step 2 point 5b. Else CHARGE must be loaded since only ACCFAM and CHARGE have VAL= SEPs.

2. CHARGE has SSJ= SEP so it can use UI = 377777B and attach PROFILO file.
   a. validate user or abort user with message if validation fails.
   b. set VAL= flag clear in UIDW via SSJ= param block
   c. there are no COA areas for charge system
   d. subsequent calls will only issue charge and project number to the account dayfile
   e. set up accounting dayfile message for charge and project number

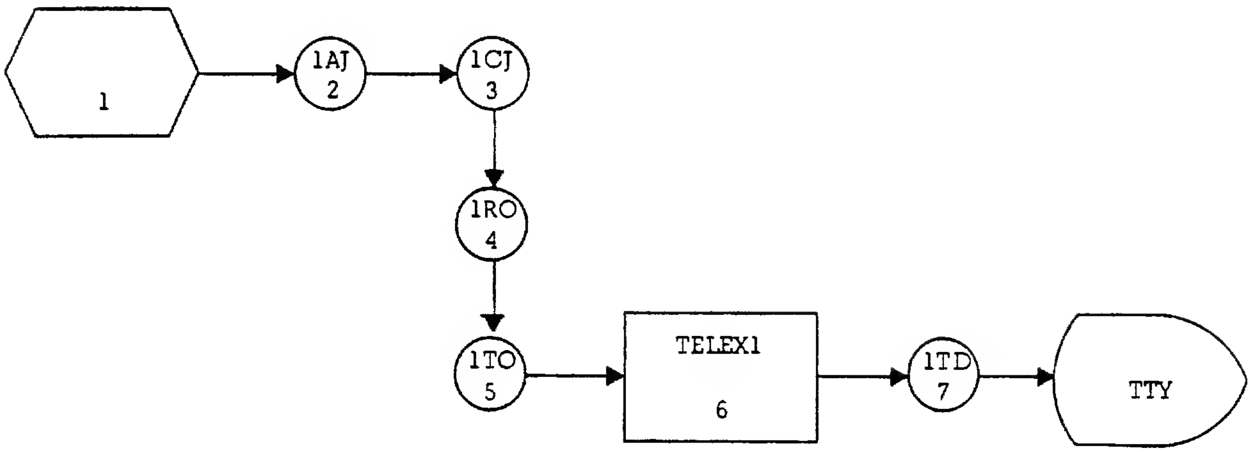
3. Special charge functions - but no change to CPA. Issue message from e. Set no charge required for TXOT origin jobs in VSTT word byte 0 bit 56 of TELEX FL.

C. VALIDATION (VAL=) (Continued)

III. SYOT does not need to be validated in order to run. (i.e., VAL=bit in UIDW is never set).

D. OUTPUT

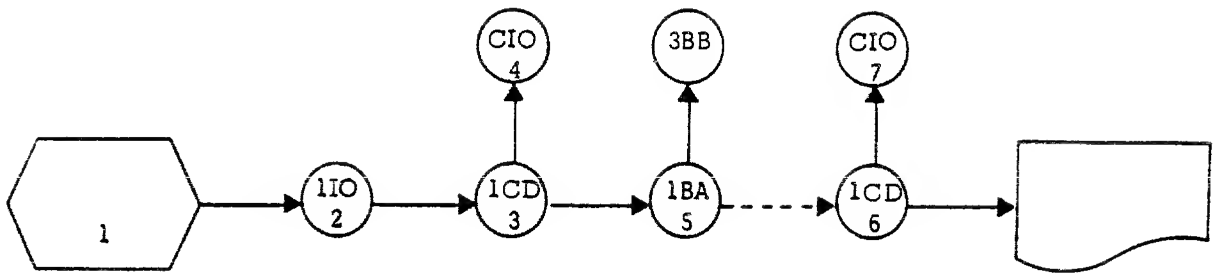
I. TELEX (TXOT)



1. Job completes
2. See Termination in ADVANCEMENT
3. 1CJ discovers that this is a TXOT type and that TELEX is active so it calls 1RO, Else see Termination in ADVANCEMENT.
4. 1RO rolls job out creates rollout file (but no FNT/FST entry) and sets terminal table entry via COMPGTN
5. 1TO transfers any output data to TELEX1 if any. (3 POTs at a time).
6. TELEX1 tells 1TD about OUTPUT if any
7. 1TD transfers OUTPUT or results of job info to user on TTY

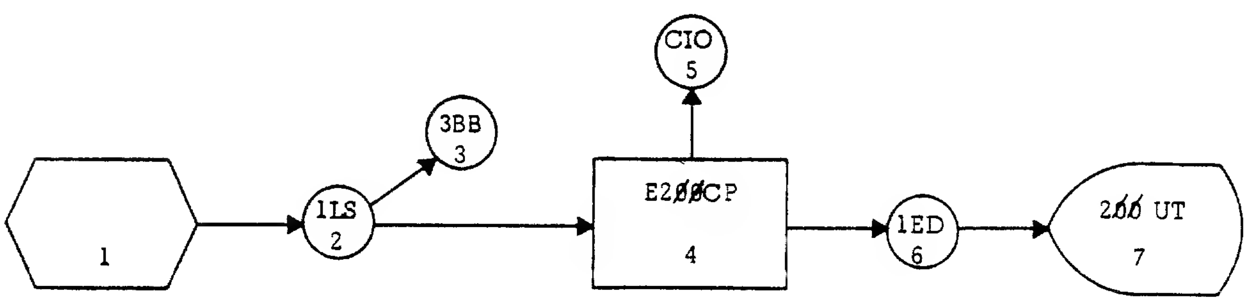
D. OUTPUT (Continued)

II. SYOT/BCOT



1. OUTPUT entry in FNT/FST of type BCOT (SYOT will be changed to PCOT by 1 1CJ)
2. Find OUTPUT queue entry with highest queue priority
3. DRQR request queues activates driver
4. Read first sector of OUTPUT file to CM buffer
5. Create banner page via 3BB.
6. Copy first and subsequent sectors from CM buffer to printer
7. Copy subsequent sectors from MS to CM buffer

III. EIOT

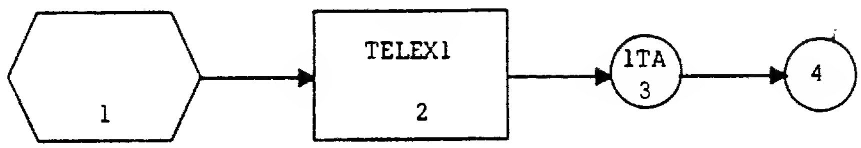


1. OUTPUT file entry in FNT/FST of type EIOT.
2. 1LS executive finds OUTPUT entry with highest queue priority
3. 3BB creates banner page in CM buffer
4. E200CP transfer data from CIO buffer to 1ED buffer
5. Copy rest of MS OUTPUT file to CIO buffer
6. Transfer data to 200 UT
7. Print output on remote printer

E. MULTI-TERMINAL (MTOT)

I. TXOT (only primary file sort jobs are defined as MTOT type as of June 1974).

1. Initiation



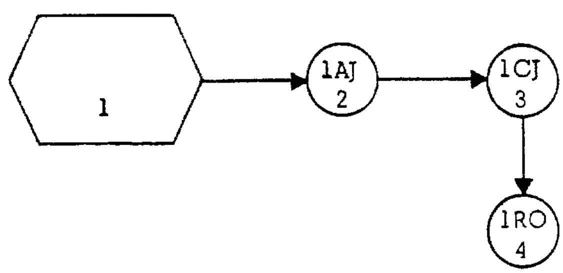
1. Terminal(s) request primary file sort either directly or indirectly

2. Set up ITAQ 1TA queue

3. a. 1TA sets up a Rollin/Rollout file of type MTOT and system sector
   b. Create FNT/FST for above
   c. Enter primary FNTs from TELEX pot.

4. Control card will be placed in CPA
   Control card buffer CSBW by 1AJ
   (3AA routine BMT) from a TELEX pot.

2. Completion



1. MTOT type job completes normally

2. 1AJ detects EOR on Control Card record

3. 1CJ finds MTOT type and finds that TELEX is active else complete job as in ADVANCEMENT Termination

4. 1RO sets terminal tables for affected terminals waiting on SORT and does not create a Rollout file - Sets error or complete status for files to be sorted in Terminal Table.

II. BCOT/EIOT/SYOT

1. MTOT type jobs are not defined.

APPENDIX C
CYBER 170 STATUS AND CONTROL REGISTERS

INTRODUCTION

NOS is designed to utilize the special hardware facilities of the CYBER 170.

In order to accomplish this end, several PP routines were modified and written to use the hardware. Since NOS can be deadstarted on any 6000 type machine, the system must be able to recognize and utilize the available hardware. In order to recognize a CYBER 170 machine, the system must determine if the Status and Control Register (SCR) exists. At deadstart, SET will status the SCR by active test on channel 16 (if active SCR exists). SET will also determine if SCR exists. During system operation, MTR will call 1MB to monitor the SCR and utilize the facilities contained therein.

The facilities include:

- Unhangable channel commands
- Unhangable PP to CM reference
- Memory parity both CM and PP
- Channel parity
- SECDED memory error detection
- Individual PP deadstart
- Double speed PPs
- Etc.

The key to using these facilities lies in the use of the SCR. All of the following information has been obtained from the following manuals:

- CYBER 170 Model 175
- Hardware Reference Manual Input/Output Specs

The glossary defines the terms used in this package.

CEJ	Central Exchange Jump
CM	Central Memory
CMC	Central Memory Control
CP	Central Processor
CPU	Central Processor Unit
CSU	Central Storage Unit
ECS	Extended Core Storage

EM	Exit Mode
FLC	Field Length for CM
FLE	Field Length for ECS
I/O	Input/Output
MA	Monitor Address
MEJ	Monitor Exchange Jump
MF	Monitor Flag
MOS	Metal Oxide Semiconductor
P	Program Address
PP	Peripheral Processor
PPM	Peripheral Processor Memory
PPS	Peripheral Processor Subsystem
RAC	Reference Address for CM
RAE	Reference Address for ECS
RNI	Read Next Instruction
SECDED	Single-Error Correction, Double Error Detection

The hardware is designed with up to 262K words of memory distributed over up to 2 chassis called Central Storage Units (CSU). Each CSU can be accessed by a Central Memory Control (CMC). A system can have only one CMC, and only two CPUs maximum. Each CMC then has 5 ports for CM access, as shown in figure C-1. These CSU units are called chassis and are placed physically in bays as shown in figure C-2 for a Model 175. The logic diagram 1 in figure C-2 shows that for any chassis to access central memory CSU, it must interface via the CMC.

Figure C-3 shows the CMC error communications.

Figure C-4 shows the central memory address format.

Figure C-5 shows the CYBER 170 DS panel.

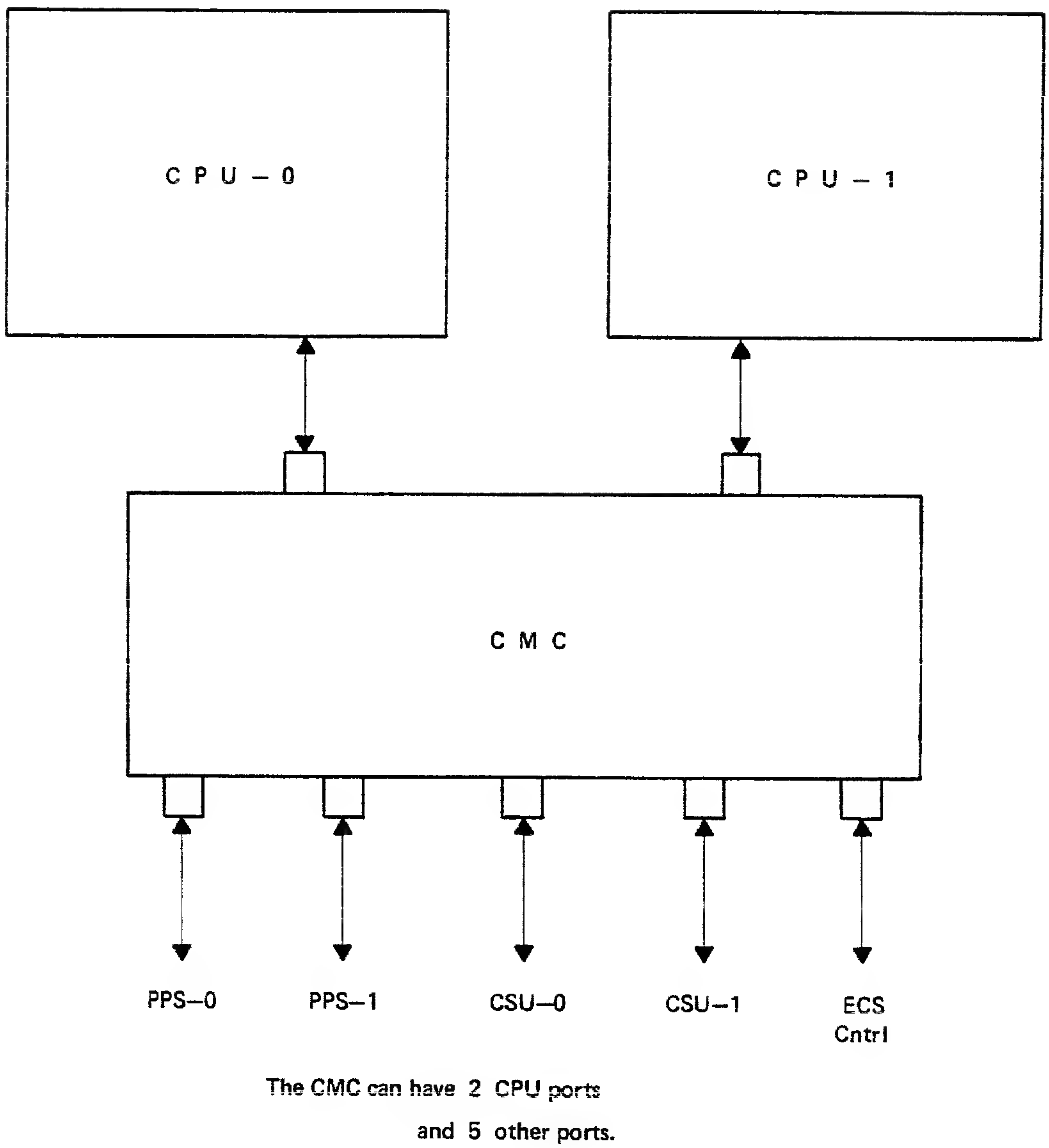


Figure C-1. CMC Ports

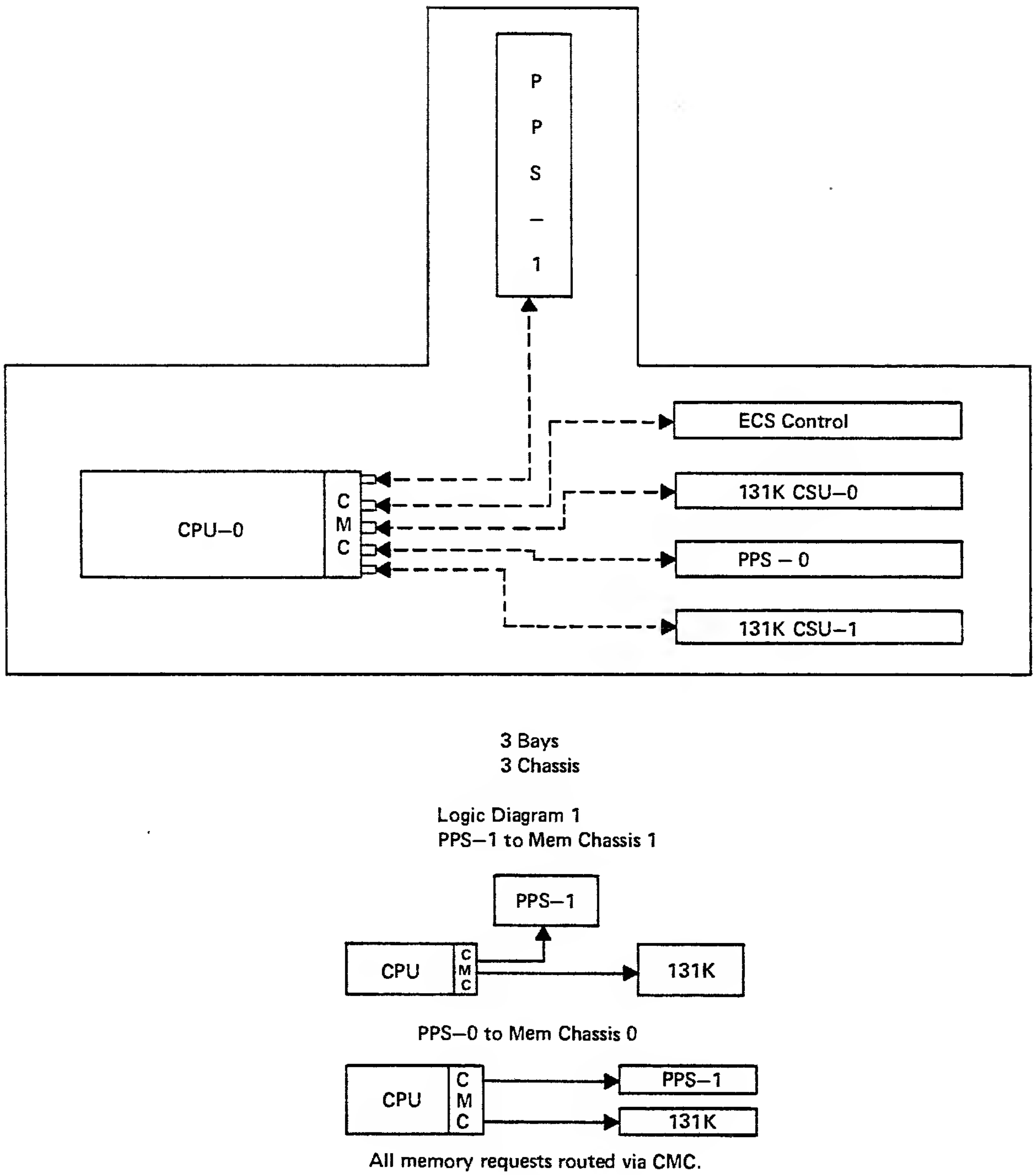


Figure C-2. CYBER 175 Configuration

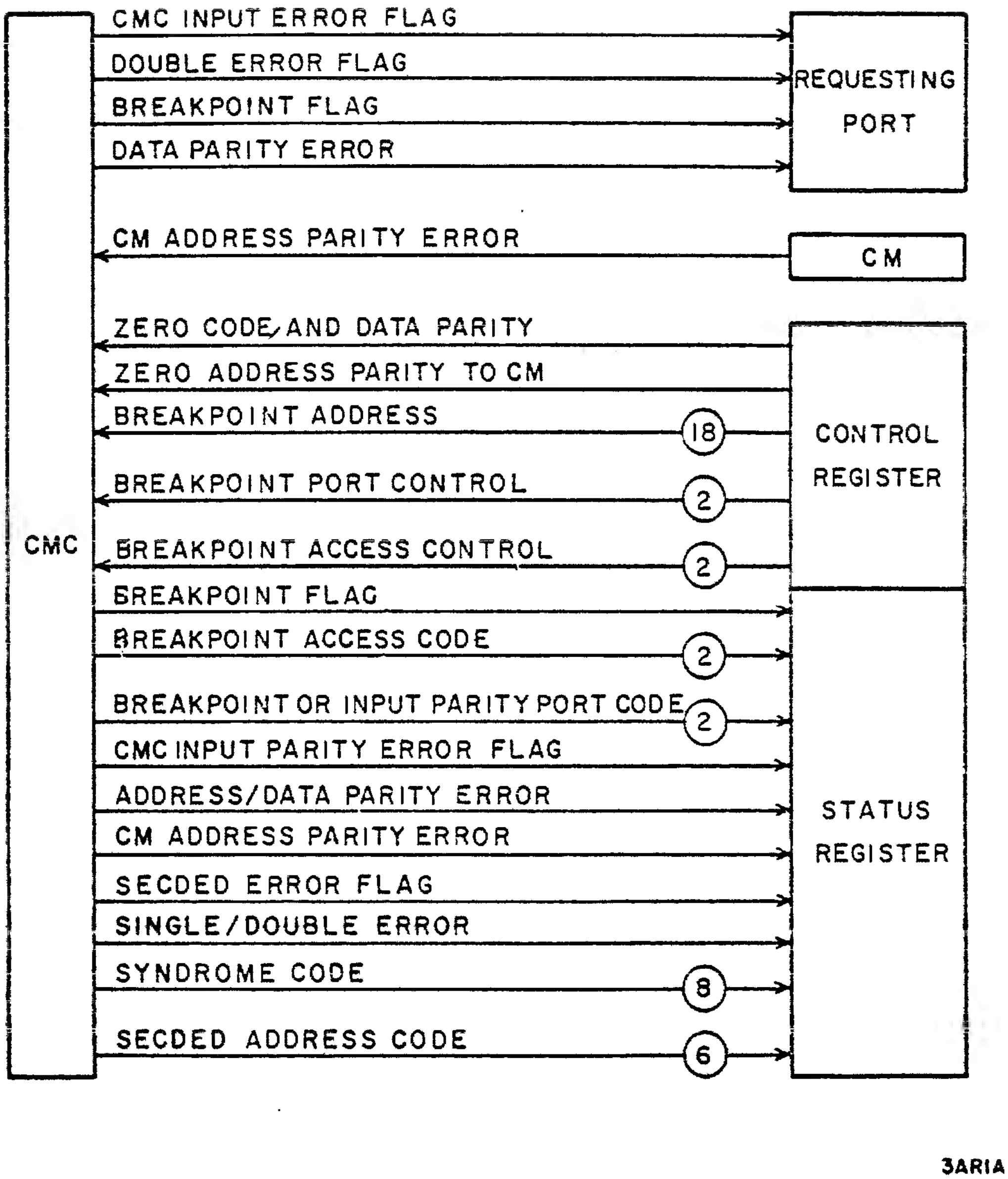


Figure C-3. CMC Error Communications

CENTRAL MEMORY

ADDRESS FORMAT

The 18-bit CM address is partially translated in CMC to a 14-bit address and 16 separate go bank signals. The translation is somewhat different for the models 172/173/174 and the model 175. On the models 172/173/174, bits 0, 1, 2, and 17 are used for bank selection. On the model 175, bits 0, 1, 2, and 3 are used for bank selection. In each case, the most significant bank select bit actually selects one of the two CSU chassis. The address formats are shown in figure C-4.

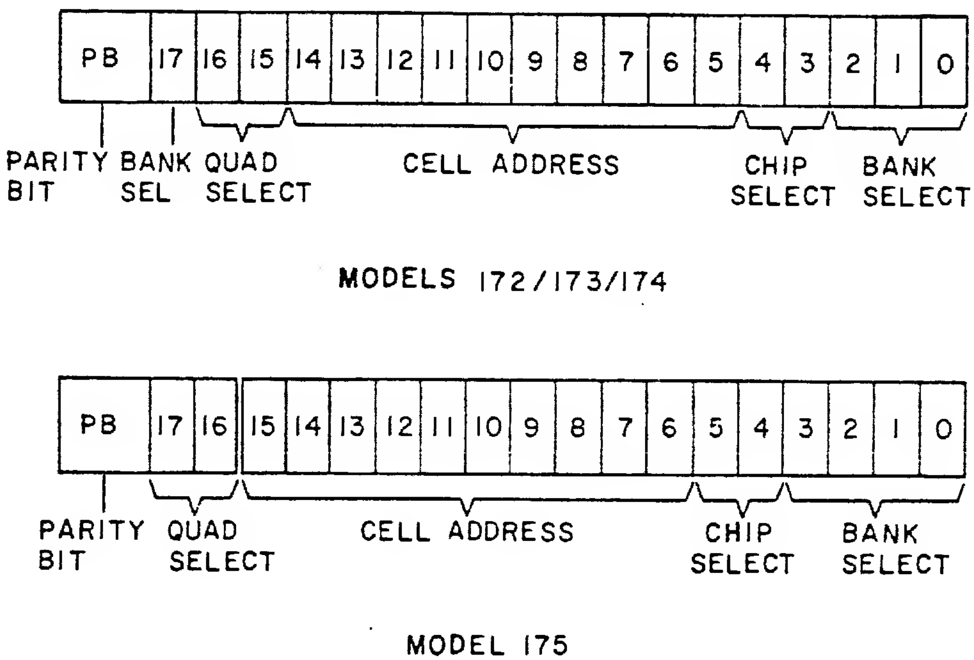
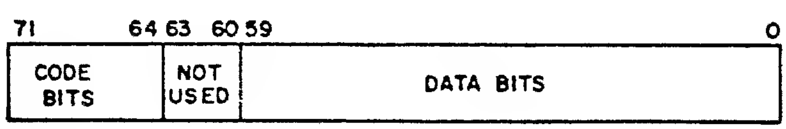


Figure C-4. Central Memory Address Formats

DATA FORMAT

Central memory is capable of sending and receiving 68 bits of information. The 68 bits are comprised of 60 bits of data plus 8 SECDED code bits which are added and checked as the data is passed through CMC. The data format is shown below.



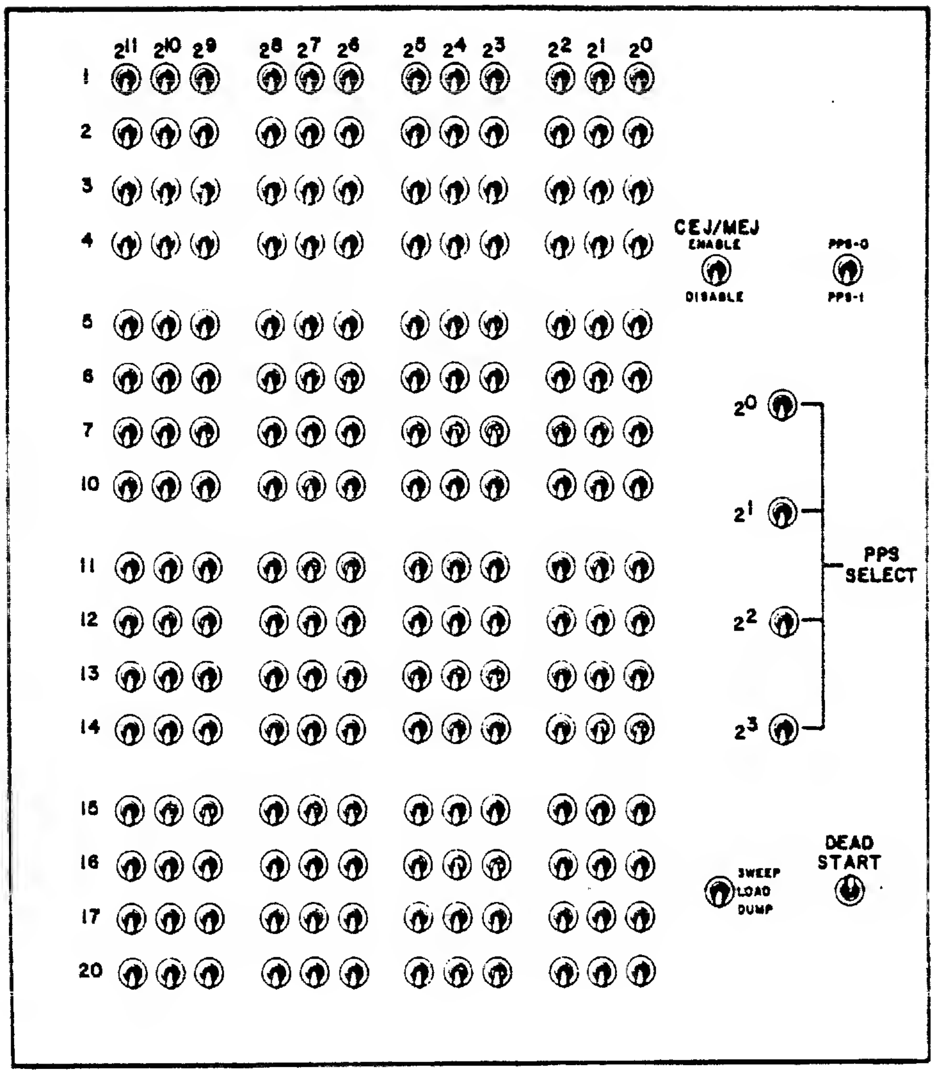


Figure C-5. CYBER 170 Deadstart Panel

STATUS AND CONTROL REGISTER

The status and control register provides control of the new features without impacting current software. It is permanently hardwired on channel 16. It has bit assignments to monitor the parity error and SECDED networks. It also is the source of control for testing these networks. Additional controls are provided for the breakpoint feature, PP speed enhancement, and maintainability features.

An additional abridged status/control register is present in a 20-pp system. It is contained on the second PPS chassis and contains only those bits that affect the additional PP's. The channel assignment is 36. There is also a bit in the prime status register that indicated a bit set someplace in the second register. This allows only one test to interrogate both registers.

Channel 16 is an internal channel and is always active. This channel has a 12-bit output register to hold a descriptor word sent from a PP. It also has a 12-bit input register to hold the status information to be read by a PP. An output will set the channel full, thus keeping any other PP from doing an output on this channel. An input must be done to clear the full after the output. This frees the channel for usage by the other PP's. To maintain consistent control of this channel, all software routines that access the status and control register channel must provide an output followed by an input.

The descriptor word sent from a PP contains the function and an address to designate the 12-bit word or single bit on which the function is to be performed.

Instruction Code	Descriptor Functions
2000 (LDC yxxx)	0xxx read
yxxx	1xxx test
7216 (OAN ch 16 or 36)	2xxx clear
7016 (IAN ch 16 or 36)	3xxx test/clear
	4xxx set
where y = function	5xxx test/set
and xxx = word or bit address	6000 clear all
	7000 test error

Description Word

Function Code	Not Used	Word or bit address
11 … 9	8	7 … 0

A read function translates xxx as the word address and selects the 12-bit word to be placed in the input register. For the other functions, xxx is translated as a bit address and selects the bit on which the function will be performed. A test function reads the bit that is selected and places it in the lowest order bit position of the input register. A set function forces the appropriate bit to a one, and a clear function forces the bit to a zero. The test/set and test/clear functions first read the selected bit and then either set or clear the bit as requested. The clear all function forces all the bits in the status and control register to a zero. A deadstart master clear also clears all the bits in the status and control register. The test error function performs a logical OR test of the lowest order 40 bits, which includes all the error flags of the status register. This allows a software routine to determine, with this single test, whether or

not an error has been recorded in the status register. Further interrogation can then be done to determine the actual error status.

Because there is no provision in the status and control register channel for writing a 12-bit word, all of the control bits must be set individually with a set function.

Light modules containing light emitting diodes (located on the PP chassis) provide a visual display of each of the status bits of the status and control register.

Programming considerations for the status and control register, channel 16 (and 36) are as follows:

Instruction	
AJM 64	Not needed because the channel is always active.
IJM 65	Not needed because the channel is always active.
IAM 71	Hangs the PP with channel empty if more than one word is input.
OAM 73	Hangs the PP with channel full if more than one word is output.
ACN 74	Hangs the PP because the channel is always active.
DCN 75	Executes, but does not disconnect the channel; becomes a two trip pass.
FAN 76	Hangs the PP because the channel is active.
FNC 77	Hangs the PP because the channel is active.

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	13	12	11	10	7	6	5	4	3	2	1	0
BIT OCTAL	11	10	9	8	7	6	5	4	3	2	1	0
S/C		S	S	S	S	S	S		S	S	S	S
FUNCTION		TE	TE	TE	TE	TE	TE		TE	TE	TE	TE
CHAN 36					X	X						X
DISPLAY		X	X	X	X	X	X		X	X	X	X
WORD												
0	Not Used	Error in 2nd PPS	CSU-1 Fault	CSU-0 Fault	Inter PPS Parity	Inter PPS Parity	CMC Parity Error	Not Used	SECDED Error	CSU-1 Address Parity Error	CSU-0 Address Parity Error	Read Pyramid Parity Error

Bit 10 (Error in 2nd PPS): Tests 0 thru 39 of PPS-1

Bit 5 (CMC Parity Error): Loads and Locks Bits 54, 55, 139, 140, 183

Bit 3 (SECDED Error): Loads and Locks Bits 40 thru 53

PPS	PP Subsystem
CSU	Central Storage Unit (Controller)
SECDED	Single Error Correction, Double Error Detection
Display	LED Display on Chassis
Chan 36	For access to 2nd bank of PPS, i.e., PPS-1 PP # 20 t 32 abbreviated SCR
S/C	Status or Control
Functions	Blank .... read, test, clear, test/clear, set test/set and clear all.
	TE .... same as Blank plus TEST ERROR part of the test error function.
	D .... same as Blank but cleared at Deadstart.
	R .... Read only!

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	23	22	21	20	19	18	17	16	15	14	13	12
BIT OCTAL	27	26	25	24	23	22	21	20	17	16	15	14
S/C	S	S	S	S	S	S	S	S	S	S	S	S
FUNCTION	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE
CHAN 36	X	X	X	X	X	X	X	X	X	X	X	X
DISPLAY	X	X	X	X	X	X	X	X	X	X	X	X
WORD												
1	PP9 Memory Parity Error	PP8	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0 Memory Parity Error	CPU-1 P Register Parity Error	CPU-0 P

RNI parity from CSU

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	35	34	33	32	31	30	29	28	27	26	25	24
BIT OCTAL	43	42	41	40	37	36	35	34	33	32	31	30
S/C	S	S	S	S	S	S	S	S	S	S	S	S
FUNCTION	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE	TE
CHAN 36	X	X	X	X	X	X	X	X	X	X	X	X
DISPLAY	X	X	X	X	X	X	X	X	X	X	X	X
WORD												
2	Channel 13 Parity Error	12	11	10	7	6	5	4	3	2	1	Channel 0 Parity Error

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	47	46	45	44	43	42	41	40	39	38	37	36
BIT OCTAL	57	56	55	54	53	52	51	50	47	46	45	44
S/C	S	S	S	S	S	S	S	S			S	S
FUNCTION	R	R	R	R	R	R	R	R			TE	TE
CHAN 36												
DISPLAY	X	X	X	X	X	X	X	X			X	X
WORD												
3	Syndrome Bit 7	6	5	4	3	2	1	Syndrome Bit 0	Not Used	Not Used	Power Shut-down imminent	Main Power Failure

Loaded and
Locked by
Bit 3
Memory SECDED Error
Clear Bit 3 unlocks these bits
See Figure C-6

Power/Environment
Abnormal Condition

The SECDED code is generated when written to CM and is appended to the 60 bit data word.

On read, the SECDED code is regenerated and a logical difference is made with the SECDED appeded to the 60 bit data word. This creates the SYNDROME codes shown below.

CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT	CODE	BIT
000	⑥	040	①	100	①	140	②	200	①	240	②	300	②	340	50
001	①	041	②	101	②	141	53	201	②	241	57	301	58	341	③
002	①	042	②	102	②	142	54	202	②	242	59	302	④	342	②
003	②	043	0	103	1	143	②	203	2	243	②	303	②	343	③
004	①	044	②	104	②	144	40	204	②	244	④	304	④	344	②
005	②	045	23	105	3	145	②	205	5	245	②	305	②	345	③
006	②	046	22	106	8	146	②	206	9	246	②	306	②	346	③
007	10	047	②	107	②	147	③	207	②	247	44	307	③	347	②
010	①	050	②	110	②	150	41	210	②	250	43	310	48	350	②
011	②	051	47	111	7	151	②	211	6	251	②	311	②	351	28
012	②	052	27	112	31	152	②	212	11	252	②	312	②	352	③
013	13	053	②	113	②	153	③	213	②	253	③	313	③	353	②
014	②	054	29	114	30	154	②	214	16	254	②	314	②	354	③
015	17	055	②	115	②	155	③	215	②	255	③	315	③	355	②
016	18	056	②	116	②	156	③	216	②	256	③	316	③	356	②
017	②	057	③	117	52	157	②	217	③	257	②	317	②	357	③
020	①	060	②	120	②	160	42	220	②	260	45	320	49	360	②
021	②	061	46	121	51	161	②	221	56	261	②	321	②	361	③
022	②	062	32	122	55	162	②	222	15	262	②	322	②	362	③
023	14	063	②	123	②	163	③	223	②	263	③	323	36	363	②
024	②	064	33	124	35	164	②	224	39	264	②	324	②	364	20
025	19	065	②	125	②	165	③	225	②	265	③	325	③	365	②
026	21	066	②	126	②	166	③	226	②	266	③	326	③	366	②
027	②	067	③	127	③	167	②	227	③	267	②	327	②	367	③
030	②	070	34	130	37	170	②	230	38	270	②	330	②	370	③
031	24	071	②	131	②	171	③	231	②	271	③	331	③	371	②
032	25	072	②	132	②	172	12	232	②	272	③	332	③	372	②
033	②	073	③	133	③	173	②	233	③	273	②	333	②	373	③
034	26	074	②	134	②	174	③	234	②	274	③	334	③	374	②
035	②	075	4	135	③	175	②	235	③	275	②	335	②	375	③
036	②	076	③	136	③	176	②	236	④	276	②	336	②	376	③
037	③	077	②	137	②	177	③	237	②	277	③	337	③	377	②

3AR38

The 8 syndrome bits along with 6 address bits associated with the memory reference are sent to the status register. This information can then be interpreted to allow determination of failing CSU, memory bank, memory quadrant, and (in the case of single correctable errors) the failing bit. This information makes it possible for the maintenance engineer to isolate the failure to a module level.

① Syndrome code bit failed (single code bit set)

② Double error or multiple double error (even no. of code bits set)

③ Multiple error reported as single error (5 or 7 code bits set)

④ Not used due to 64-bit algorithm

5 Syndrome codes above are octal representations of 8 syndrome code bits

⑥ No error was detected

Figure C-6. SECDED Syndrome Codes/Corrected Bits

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	59	58	57	56	55	54	53	52	51	50	49	48
BIT OCTAL	73	72	71	70	67	66	65	64	63	62	61	60
S/C	S	S	S	S	S	S	S	S	S	S	S	S
FUNCTION	R	R	R	R	R	R	R	R	R	R	R	R
CHAN 36												
DISPLAY	X	X	X	X	X	X	X	X	X	X	X	X
WORD												
4	Bit 1	Break Point Function Code Bit 0	Bit 1	Break Point Port Code Bit 0	Bit 1	Parity Error Port Code Bit 0	Syn-drome Chassis Bit	Bit 1	Syn-drome Quad Bit 0	Bit 2	Bit 1	Syn-drome Bank Bit 0
	Loaded and Locked by Bit 77				From CMC to identify port Loaded and Locked by Bit 5		Loaded and Locked by Bit 3					

CMC .... Central Memory Control

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	71	70	69	68	67	66	65	64	63	62	61	60
BIT OCTAL	107	106	105	104	103	102	101	100	77	76	75	74
S/C	S	S	S	S	S	S	S	S	S	S	S	S
FUNCTION	R	R	R	R	R	R	R	R	R	R	R	R
CHAN 36	X	X	X	X	X	X	X	X	X	X	X	X
DISPLAY	X	X	X	X	X	X	X	X	X	X	X	X
WORD												
5	11	10	9	8	7	6	5	4	3	2	1	PPS P Register Bit 0

If Bit 83 Clear, bits 60 thru 71 display P register for the PP selected by bits 120 thru 123 and bits 72 thru 75 display the PP selected.

Set, the contents of the P register is latched (i.e., locked) and retained on every CM breakpoint HIT.

Set, end bit 76 gets set, bits 60 thru 75 ere held unit bit 76 is cleared.

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	83	82	81	80	79	78	77	76	75	74	73	72
BIT OCTAL	123	122	121	120	117	116	115	114	113	112	111	110
S/C	C		C	C	C	C	S	S	S	S	S	S
FUNCTION	D		D	D					R	R	R	R
CHAN 36	X		X	X	X	X			X	X	X	X
DISPLAY												
WORD												
6	PPS Break Point Mode Select	Not Used	Force Zero Parity on all PP Memories	Force Zero Parity on all Channels	Set C5 full	Clear Central Memory Busy	CMC Break Point Match	CMC Break Point HIT	3	2	1	PP Code Bit 0

Same comments as bit 60 thru 71.

One shop operation, i.e., set bit 79 set C5 full. This allows a PP hung on an unanswered CM request to react as if C5 went full (received data from CM0 and continue. This allows recovery from a lost accept signal on a PP to CM request.

Busy FF (flip flop) in PPS. One shot operation, i.e., set bit 78 clears the interlock Central Memory Busy.

Loads and Locks bits 56 thru 59.

Same comments as bits 60 thru 71.

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	95	94	93	92	91	90	89	88	87	86	85	84
BIT OCTAL	137	136	135	134	133	132	131	130	127	126	125	124
S/C	C										C	C
FUNCTION	D										D	D
CHAN 36	X											
DISPLAY	X											X
WORD												
7	Stop on PP Memory Parity Error	N O T U S E D									Inhibit CMC Request	All PP's 100 ns Major Cycle

Stop the PP which received a memory parity error.

Set means 2Xspeed

Clear means 1Xspeed (=1000 ns)

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	107	106	105	104	103	102	101	100	99	98	97	96
BIT OCTAL	153	152	151	150	147	146	145	144	143	142	141	140
S/C	C	C	C	C	C	C	C	C	C	C	C	C
FUNCTION												
CHAN 36												
DISPLAY												
WORD												
10	11	10	9	8	7	6	5	4	3	2	1	Break Point Address Bit 0

Absolute 18 bit address sent to CMC for Breakpoint condition.

Bits 96 thru 113 are used by CMC for Breakpoint address when the condition code is set in bits 116 and 117.

See Figure C-7.

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975 10

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	119	118	117	116	115	114	113	112	111	110	109	108
BIT OCTAL	167	166	165	164	163	162	161	160	157	156	155	154
S/C		C	C	C	C	C	C	C	C	C	C	C
FUNCTION												
CHAN 36												
DISPLAY		X										
WORD												
11	Not Used	Inhibit Single Error Report	21	20	19	Break Point Condition Code 18	17	16	15	14	13	Break Point Address 12

Bits 117–114: See diagram (P. 2-15)

Bits 113–108: Same comments as bits 96 through 107

Single errors are not recorded (when detected) in SCR when this bit 118 is set.

BREAKPOINT CHECK FOR DIAGNOSTICS NOT CURRENTLY SUPPORTED BY NOS

CMC performs a breakpoint check on reference to CM when breakpoint is selected. Breakpoint is selected from the status and control register in the PPS.

CMC receives 18 breakpcint address bits, two port control bits, and two access control bits. Table C-1 lists the breakpoint control translations.

TABLE C-1. BREAKPOINT CONTROL TRANSLATION

Control Bit				Translation
117	116	115	114	
0	0	X	X	Breakpoint Check Disabled
0	1	X	X	Breakpoint Check for PP Ports
1	0	X	X	Breakpoint Check for CPU Ports
1	1	X	X	Breakpoint Check for PP and CPU Ports
X	X	0	0	Breakpoint Check on Read
X	X	0	1	Breakpoint Check on Write
X	X	1	0	Breakpoint Check on RNI
X	X	1	1	Breakpoint Check on any access

The 18-bit address of each CM reference is compared to the breakpoint address bits. If there is a match and if the requesting unit is selected by the port control bits, and if the type of access is one that is selected by the access control bits, the breakpoint flag is sent to the requesting unit.

The breakpoint flag is also sent to the status register along with the two port code bits and the two access code bits.

When executing an exchange jump, this operation is treated by breakpoint as both a read and a write. A return jump is treated as a write.

BREAKPOINT NOTES FOR MODEL 175

1. Since breakpoint is for an address request to CM, a breakpoint does not occur for an instruction executed from the instruction stack if the instruction entered the instruction stack before selecting breakpoint.

2. The value of P plus RAC when the CPU is stopped by breakpoint may not correspond with the value of the breakpoint address because the CPU normally requests two words ahead of P on an RNI.

3. The value of P plus RAC when the CPU stops for a breakpoint on an increment address may not correspond with the value of the P+RAC of the increment instruction. Advancing P is based on the 60-bit word of instructions entering CIW instead of any given parcel of CIW being executed.

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975 11

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	131	130	129	128	127	126	125	124	123	122	121	120
BIT OCTAL	203	202	201	200	177	176	175	174	173	172	171	170
S/C			C	C	C	C	C	C	C	C	C	C
FUNCTION					D	D	D	D	D	D	D	D
CHAN 36					X	X	X	X	X	X	X	X
DISPLAY									X	X	X	X
WORD												
12	Not Used	Not Used	Force Zero Address Parity CMC to CM	Zero Data Code and Parity CMC to CM	CSU CMC CPU Master Clear	Force PP Dead Start	Force Exit on Selected PP	PP Select Auto/ Manual Mode	3	2	1	PP Select Code Bit 0

Zero parity bit

PP remains in DS condition until the bit 126 is cleared, i.e. hang on input on its associated channel

One shot operation The selected PP will complete the current instruction and go on to the next without waiting for conditional replies.

Select 1 of 10 PP's for forced

EXIT	bit 125
DEADSTART	bit 126
DISPLAY	bit 124

Manual bits 72-75 display PP selected by switches on chassis
Auto bits 72-75 display PP selected by bits 120-123

97404700D

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	143	142	141	140	139	138	137	136	135	134	133	132
BIT OCTAL	217	216	215	214	213	212	211	210	207	206	205	204
S/C	C	C	C	S	S	S	S	S	C	C	C	C
FUNCTION												
CHAN 36												
DISPLAY				X	X							
WORD												
13	Clock Frequency Margins 2	1	0	ECS Transfer Error	CMC Address Parity Error Type	ECS 2	Error 1	Status Bit 0	Refresh Fast	Margins Slow	ECS coupler Zero parity Code 1	Code 0

Bits 141 thru 143 are code bits for selecting clock margins for Master clock

Bits 139 and 140 loaded and locked by Bit 5

Memory Refresh Times

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	155	154	153	152	151	150	149	148	147	146	145	144
BIT OCTAL	233	232	231	230	227	226	225	224	223	222	221	220
S/C	C	C	C	C	S	S	S	S	S	S	S	S
FUNCTION												
CHAN 36												
DISPLAY					X	X	X	X	X	X	X	X
WORD												
14	Select All/One RVM	Hi/Lo RVM	Clock Width Wide	Pulse Narrow	RVM 7	Address 6	5	4	Status 3	2	1	Bit 0

CPU Clock (bits 9–8)

Indicates Module having Reference Voltage Margins (RVM) applied. (bits 7–0)

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	167	166	165	164	163	162	161	160	159	158	157	156
BIT OCTAL	247	246	245	244	243	242	241	240	237	236	235	234
S/C	C	C	C	C	C	C	C	C	C	C	C	C
FUNCTION												
CHAN 36												
DISPLAY												
WORD												
15	RVM		QUADRANT		NUMBER	SELECT						
	11	10	9	8	7	6	5	4	3	2	1	0

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

Register 16

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	179	178	177	176	175	174	173	172	171	170	169	168
BIT OCTAL	263	262	261	260	257	256	255	254	253	252	251	250
S/C	C	C	C	C	C	C	C	C	C	C	C	C
FUNCTION												
CHAN 36					X	X						
DISPLAY												
WORD												
	MEM RECONFIGURATION BIT				PPS TO CMC ZERO PARITY BIT		RVM MODULE ADDRESS BIT					
	3	2	1	0	DATA	ADRS	5	4	3	2	1	0

Bits 176 and 183 correspond to memory degrad switches on Figure C-7.

Zero parity bit on Data or address specified by the PP

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	191	190	189	188	187	186	185	184	183	182	181	180
BIT OCTAL	277	276	275	274	273	272	271	270	267	266	265	264
S/C			C	C	C	C	C	C	C	C	C	C
FUNCTION												
CHAN 36			X	X								
DISPLAY									X			
WORD												
17	Not Used	Not Used	Software Lock Clear	Software Lock Test	Zero Parity CPU-1 to CMC	Zero Parity CPU-0 to CMC	Zero Parity CPU-1 to CMC	Zero Parity CPU-0 to CMC	Double Error From CMC	Memory Configuration Bit 6	Memory Configuration Bit 5	Memory Configuration Bit 4
			Diagnostic Aids		Zero parity bit on data or address				Indicates double error from SECDED. See bit 5	Same as comment on bits 176-179		

Memory Size	Range of Address	Normal Operation: Adrs Range Control Sw 1 2 3 4	Normal Operation: Bad Quad Code Sw 5 6 7	Degraded Operation: Bad Quadrant		Degraded Operation: Switches 1 2 3 4	Degraded Operation: Switches 5 6 7
262 K	0-777777	1 1 1 1	1 1 1	CSU-0	0 0	1 1 1 0	0 0 0
					0 1		0 0 1
					1 0		0 1 0
					1 1		0 1 1
				CSU-1	0 0		1 0 0
					0 1		1 0 1
					1 0		1 1 0
					1 1		1 1 1
196 K	0-577777	1 1 1 0	1 1 1	CSU-0	0 0	0 1 1 1	0 0 0
					0 1		0 0 1
					1 0		0 1 0
					1 1		0 1 1
				CSU-1	0 0		1 0 0
					0 1		1 0 1
131 K	0-377777	0 1 1 1	1 1 1		0 0	0 1 1 0	0 0 0
					0 1		0 0 1
					1 0		0 1 0
					1 1		0 1 1
98 K	0-277777	0 1 1 0	1 1 1		0 0	0 0 1 1	0 0 0
					0 1		0 0 1
					1 0		0 1 0
65 K	0-177777	0 0 1 1	1 1 1		0 0	0 0 1 1	0 0 0
					0 1		0 0 1
49 K	0-137777	0 0 1 0	1 1 1		0 0	0 0 0 1	0 0 0
					0 1		0 0 1
32 K	0-077777	0 0 0 1	1 1 1			NO DEGRADE	

Figure C-7. Memory Selection Scheme for Model 172/173/174

CYBER 170 STATUS AND CONTROL REGISTERS (SCR) AS OF JANUARY 1975

BIT IN WORD	11	10	9	8	7	6	5	4	3	2	1	0
BIT DECIMAL	203	202	201	200	199	198	197	196	195	194	193	192
BIT OCTAL	313	312	311	310	307	306	305	304	303	302	301	300
S/C			S	S	S	S	S	S	S	S	S	S
FUNCTION			R	R	R	R	R	R	R	R	R	R
CHAN 36												
DISPLAY								X	X	X	X	X
WORD												
20	Not Used	Not Used	PPM 4	3	2	Reconfiguration 1	0	Monitor Flag Status 1	Monitor Flag Status 0	ECS in Progress Flag	CPU-1 Stopped	CPU-0 Stopped

Bits 9–5: Which physical PP is logically PPO, i.e. the PPO select switches from the DS panel

Bits 4–3: Indicate CPU status monitor or program mode

Bits 1–0: Program stop

CPU EXIT MODE/ERROR RESPONSE

Since memory, all channels, and ECS now have parity, the system needs to be able to take non-default action on parity conditions. In order to implement the non-default action, three bits have been added to the EM portion of the exchange package (EP).

Figure C-8 shows the EP

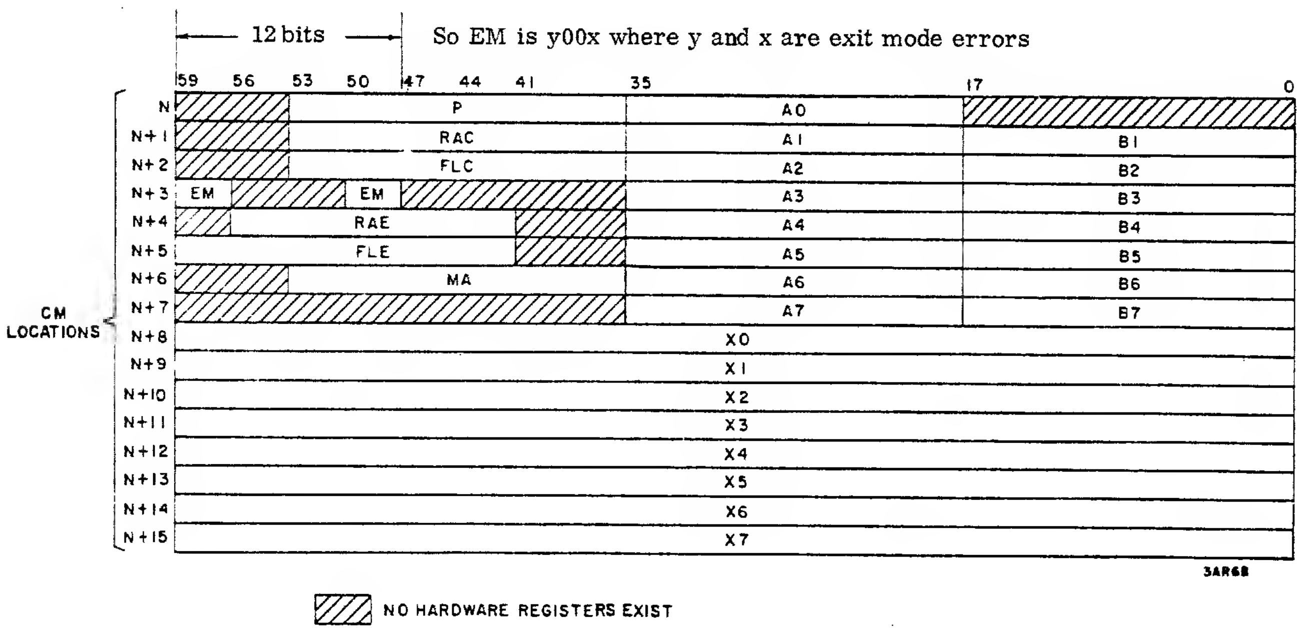


Figure C-8. Hardware Exchange Package

When the CPU detects or is informed of an error, it will record the error. Depending on the type of error and the mode selection bits, the program in execution may be interrupted. If the error is an illegal instruction, breakpoint, or address range errors on RNI or branch, the program interruption will be unconditional. For other types of errors, the mode selection bits determine whether or not the program will be interrupted. If the mode selection bit set and the corresponding condition is detected, the program will be interrupted. These sections are contained in word N+3 of the exchange package and are selected as follows:

Condition Bit	Mode Selection Bit	
48	48	Address range error
49	49	Infinite mode
50	50	Indefinite mode
51	57	Parity error on ECS flag register operation
52	58	CMC input error
53	59	CM data error

Error conditions 48, 49, and 50 are detected in the CPU and conditions 51, 52, and 53 are flags sent to the CPU from the CMC. The data parity error flag indicates a transmission error (or memory error in default mode) on data sent to the CPU. The CMC input error flag indicates that the address or data sent by the CPU had incorrect parity at the CMC or CM. The double error flag indicates that the SECDED network detected a double error on data that was requested by the CPU.

Any error condition detected after an exchange jump instruction has started execution is treated as an error for the program being exchanged in. Tables C-2 through C-4 explain what happens when the various kinds of errors occur.

Each of these tables lists the same error conditions. The error response is dependent on the setting of the MEJ/CEJ switch and the state of the monitor (MF) flag. The table headings specify the three combinations.

TABLE C-2. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF SET

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Illegal instruction	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.
Exit condition bit 48 set by an increment read of an address out of range	1. Read all zeros to the selected X register. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Read all zeros to the selected X register. 2. Continue execution.
Exit condition bit 48 set by an increment write of an address out of range.	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Block write operation, contents of CM is unchanged. 2. Continue execution.
Exit condition bit 48 set on RNI or branch out of range	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.
Exit condition bit 48 set on CMU instruction a. C1 or C2 > 9 b. K1 or K2 address out of range	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Stop CPU. 3. Store P and exit condition at RAC. 4. Clear P.	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Continue with next 60-bit instruction.

TABLE C-2. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF SET (Continued)

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Exit condition bit 48 set by an ECS address range check	1. Force ECS instruction to execute as a pass instruction. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Force ECS instruction to execute as a pass instruction. 2. Exit to next 60-bit word. 3. Continue execution with next 60-bit word.
Infinite condition (bit 49) Indefinite condition (bit 50) ECS flag register parity (bit 51) CMC input error condition (bit 52) CM data error condition (bit 53)	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.	1. Continue execution.
CMC input error condition (bit 52)	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Block write operation, contents of CM is unchanged. 2. Continue execution.
00 instruction	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.
Breakpoint signal from CMC (refer to breakpoint notes)	1. Execute remaining parcels of 60-bit word currently executing. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Execute remaining parcels of 60-bit word currently executing. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.

TABLE C-3. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF CLEAR

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Illegal instruction	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.
Exit condition bit 48 set by an increment read of an address out of range	1. Read all zeros to the selected X register. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Read all zeros to the selected X register. 2. Continue execution.
Exit condition bit 48 set due to an increment write of an address out of range	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Block write operation, contents of CM is unchanged. 2. Continue execution.
Exit condition bit 48 set due to an RNI or branch address out of range	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P. 4. Exchange jump to MA and set MF.	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P. 4. Exchange jump to MA and set MF.

TABLE C-3. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF CLEAR (Continued)

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Exit condition bit 48 set on CMU instruction a. C1 or C2 > 9 b. K1 or K2 address out of range	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Stop CPU. 3. Store P and exit condition at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Continue with next 60-bit instruction.
Exit condition bit 48 set by an ECS address range check	1. Force ECS instruction to execute as a pass instruction. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Force ECS instruction to execute as a pass instruction. 2. Continue execution with next 60-bit word.
Infinite condition (bit 49) Indefinite condition (bit 50) CMC input error condition (bit 52) CM data error condition (bit 53)	1. Stop CPU. 2. Store P and exist condition bits at RAC. 3. Clear P. 4. Exchange jump to MA and set MF.	1. Continue execution.
CMC input error condition (bit 52)	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Block write operation, contents of CM is unchanged.

TABLE C-3. ERROR RESPONSE WITH MEJ/CEJ ENABLED, MF CLEAR (Continued)

Error Condition	Error Response Exit Mode Selected	Error Response Exit Mode Not Selected
00 instruction	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P. 4. Exchange jump to MA and set MF.	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P. 4. Exchange jump to MA and set MF.
Breakpoint signal from CMC (refer to breakpoint notes)	1. Execute remaining parcels of 60-bit word currently executing. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.	1. Execute remaining parcels of 60-bit word currently executing. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P. 5. Exchange jump to MA and set MF.

TABLE C-4. ERROR RESPONSE WITH MEJ/CEJ DISABLED

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Illegal instruction	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Execute the illegal instruction as if it were a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.
Exit condition bit 48 set by an increment read of an address out of range	1. Read all zeros to the selected X register. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Read all zeros to the selected X register. 2. Continue execution.
Exit condition bit 48 set by an increment write of an address out of range	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Block write operation, contents of CM is unchanged. 2. Continue execution.
Exit condition bit 48 set due to an RNI or branch address out of range	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.	1. Stop CPU.
Exit condition bit 48 set on CMU instruction a. C1 or C2 > 9 b. K1 or K2 address out of range	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Stop CPU. 3. Store P and exit condition at RAC. 4. Clear P.	1. Condition (a) causes instruction to execute as pass. Condition (b) causes instruction moves or compares up to the point of address out of range. 2. Continue with next 60-bit instruction.

TABLE C-4. ERROR RESPONSE WITH MEJ/CEJ DISABLED (Continued)

Error Condition	Error Response	
	Exit Mode Selected	Exit Mode Not Selected
Exit condition bit 48 set by ECS address range check	1. Force ECS instruction to execute as a pass. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Force ECS instruction to execute as a pass. 2. Continue execution with next 60-bit word.
Infinite condition (bit 49) Indefinite condition (bit 50) ECS flag register parity (bit 51) CMC input error condition (bit 52) CM data error condition (bit 53)	1. Stop CPU. 2. Store P and exit condition bits at RAC. 3. Clear P.	1. Continue execution.
CMC input error condition (bit 52)	1. Block write operation, contents of CM is unchanged. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Block write operation, contents of CM is unchanged. 2. Continue execution.
00 instruction	1. Stop CPU.	1. Stop CPU.
Breakpoint signal from CMC (refer to breakpoint notes)	1. Execute remaining parcels of 60-bit instruction word. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.	1. Execute remaining parcels of 60-bit instruction word. 2. Stop CPU. 3. Store P and exit condition bits at RAC. 4. Clear P.

INDEX

CUT ON THIS LINE

COMMENT SHEET

CONTROL DATA CORPORATION

TITLE: KRONOS 2.1 Workshop Manual

PUBLICATION NO: 97404700 **REVISION:** D

This form is not intended to be used as an order blank. Control Data Corporation solicits your comments about this manual with a view to improving its usefulness in later editions.

Applications for which you use this manual.

Do you find it adequate for your purpose?

What improvements to this manual do you recommend to better serve your purpose?

Note specific errors discovered (please include page number reference).

General comments:

FROM: NAME: ______________________ POSITION: ______________________

COMPANY NAME: ______________________

ADDRESS: ______________________

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD

FIRST CLASS
PERMIT NO. 8241

MINNEAPOLIS, MINN.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

CUT ALONG LINE

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
P.O. Box O
8100 34th Avenue South
Minneapolis, Minnesota 55440

ATTN: Charlotte A. Cox
HQSO5C

FOLD

FOLD